

Article

An Integrated Credit-Based Incentive Protocol for Symbol-Level Network-Coded Cooperative Content Distribution among Vehicular Nodes

Ing-Chau Chang ^{1,*} , Chin-En Yen ² and Jacky Lo ¹

¹ Department of Computer Science and Information Engineering, National Changhua University of Education, Changhua 50007, Taiwan; f91jackyex@gmail.com

² Department of Early Childhood Development and Education, Chaoyang University of Technology, Taichung 41349, Taiwan; ceyen@cyut.edu.tw

* Correspondence: icchang@cc.ncue.edu.tw; Tel.: +886-4-723-2105

Received: 15 September 2018; Accepted: 18 October 2018; Published: 24 October 2018



Abstract: In traditional symbol-level network coding (SLNC)-based cooperative content distribution approaches, they ignore nodes in the vehicular ad hoc network (VANET) having various network-coded content pieces and distinct levels of interests and selfishness for different kinds of content data, which further prevents these vehicular nodes from forwarding their content information to other nodes. With these approaches, these nodes suffer from the low ratio and the long latency to receive all content information. In this paper, based on distinct levels of node interests and selfishness on different content information, we first categorize vehicular nodes into four classes, that is, the destination, intermediate, irrelevant and overhearing ones and then designate their associated credit-based incentive approaches. Second, we modify the flow of traditional SLNC-based cooperative content distribution operations and propose the content bitmap to realize the difference of network-coded content pieces among vehicular nodes. Further, we rigidly combine the proposed credit-based incentive approach with the modified SLNC-based cooperative content distribution operations in SocialCode to encourage all classes of vehicular nodes to rise their incentives for sharing content data in the cooperative content distribution process. Finally, we perform NS-2 simulations on a street map of downtown Taipei, Taiwan to exhibit the high efficiency of SocialCode over related credit-based incentive approaches by analyzing the following performance metrics, that is, average decoding percentage, file downloading delay and credits, with respect to different file sizes and total numbers of vehicular nodes. As the best knowledge we have, SocialCode is one of the first few researches that works on the integration between the credit-based incentive protocol and the SLNC-based cooperative content distribution.

Keywords: vehicular ad hoc network (VANET); cooperative content distribution; credit-based incentive protocol; levels of selfishness and interests; symbol-level network coding; SocialCode

1. Introduction

As the dedicated short-range communication (DSRC) [1,2], IEEE 802.11p [3] and IEEE 1609 WAVE architecture [4] become prominent in these years, the extension of the mobile ad-hoc network (MANET), that is, the vehicular ad hoc network (VANET), focuses on providing a dynamically formed MANET for vehicles to transfer information between each other and the road side unit (RSU). VANET has two communication scenarios, that is, Vehicle-to-Infrastructure (V2I) and Vehicle-to-Vehicle (V2V), [5]. The cooperative content distribution (CCD) is one of promising VANET applications [6,7]. Operations of CCD are explained in the following. First, the RSU intends to broadcast multimedia content data,

like electronic advertisements or coupons, through VANET to on board units (OBU) of vehicles in a given geographical area of interest (AoI). Because the contact duration between the RSU and the fast-moving vehicular node is short, the vehicular node may only receive parts of content data. Hence, the vehicular node uses several kinds of approaches to forward its content pieces to all nearby nodes when it disconnects from the RSU. First, the P2P approach is proposed to exchange received content pieces [8–10]. However, it incurs huge overhead due to high mobility and poor propagation condition in the vehicular environment. For overcoming this problem, the network coding (NC) technology is then used for CCD [11–21]. It encodes original packets at every source node, which then floods these NC-coded packets to nearby nodes. Whenever the nearby node has received one NC-coded packet, it must verify linear independence of all received packets. After receiving a certain number of linearly independent packets, the receiver can solve them to reconstruct original packets. With the merit of NC, vehicular nodes in AoI can effectively collect all content data to increase throughput, reliability and security in communication and further reduce duplicate transmissions. In Reference [13], nodes compare each other to find out which node has received most file blocks and let this node broadcasts its blocks. If this node holds the transmission opportunity too long, this approach suffers from degraded network performance when VANET topology changes rapidly. The pull-based approach is used in References [14,15]. Each node first exchanges its block availability to others and then a node can pull blocks from others. Because the node only issues its blocks after receiving the content request from its neighbor, network bandwidth is wasted due to this passive pulling. In Reference [16], after the node receiving a coded packet, it defers transmission of the re-coded packet until a fixed timer expires. There is no transmission scheduling for nodes, which may introduce serious packet collisions.

Recently, the symbol-level network coding (SLNC) technique has been proposed to support robust transmission and stimulates more active simultaneous transmissions [17]. For alleviating the aforementioned problems, the SLNC-based CCD approach, for example, CodeOn [18], CodePlay [19] and so forth, has been proposed [17–21]. In every control time slot, each vehicle broadcasts the reception status of content data in a safety message to its neighbors. For coordinating the broadcast among vehicles, these SLNC-based CCD approaches adopt the prioritized relay selection scheme to calculate the value of utility, which is defined as the number of valuable SLNC-coded content pieces that a node contributes to its neighbors through its broadcast. However, these approaches do not differentiate content pieces of a node from those of its neighbors. If each node owns the same number of content pieces, these SLNC-based CCD approaches will terminate their operations because the utility value between any two nodes is zero, in spite of some content pieces are distinct indeed.

Furthermore, these SLNC-based CCD approaches ignore nodes in VANET having distinct levels of interests and selfishness for different kinds of content data, which then prevents these vehicular nodes from forwarding their content information to other nodes. As a result, they suffer from much lower ratio and much higher latency for the vehicular node to receive all content data successfully. According to a survey in Reference [22], as the ratio of selfish node rises from 0 to 40%, the number of packet losses increases 500%. For alleviating this defect, three types, that is, credit-based [23–31], reputation-based [32,33] and tit-for-tat (TFT)-based [34,35], of incentive schemes have been proposed to encourage the node to share their contents. In this paper, we focus on the credit-based incentive scheme for stimulating the vehicular node to re-broadcast the content data it received to its neighbors by paying some credits to it. The credit-based schemes introduce the concept of credits to regulate the packet forwarding process among nodes. For each packet-forwarding operation, an authority would charge the sending node of the packet a unit of credit and pay it to each relay node. It is well known that the central control system (CCS) to pay the credit among nodes becomes the major problem if the CCS fails [23–26]. Moreover, each credit-based work has defects on its own incentive mechanism. For example, RCM [25] suffers the overspending problem, which means that it cannot estimate the number of forwarding nodes ahead such that it fails to limit whole payments. FRAME [26] may pay too fewer rewards to a node in some cases, which lowers the incentive of the node to forward packets. Most importantly, these credit-based incentive protocols work independently with the SLNC-based

CCD approach, which means rare credit-based incentive protocol can stimulate all kinds of nodes to accelerate the SLNC-based CCD process. Based on our prior work [36], we propose SocialCode (SC) that has subsequent contributions.

- (1) To solve the aforementioned defect that traditional SLNC-based CCD schemes do not realize the difference of network-coded content pieces among vehicular nodes, SC designs the content bitmap to store which network-coded content pieces every node holds. Whenever SC begins its CCD operations, every node broadcasts the beacon message to interchange its content bitmap with its neighbors. In this way, each node is able to compute the number of distinct network-coded content pieces, the re-broadcast priority and the number of shared SLNC content pieces after re-encoding.
- (2) Based on distinct levels of interests and selfishness on different content data, SC first categorizes the vehicular node into four classes, that is, the destination, intermediate, irrelevant and overhearing ones and then designates their associated network-coding and re-broadcasting operations to share SLNC content pieces efficiently among these four types of nodes.
- (3) SC improves the conventional SLNC-based cooperative content distribution protocol as mentioned above. It further rigidly combines with the proposed credit-based incentive approach to encourage all classes of vehicular nodes to rise their incentives for sharing content data. In this paper, SC proposes its distributed flow to exchange credits among four types of nodes when they re-broadcast or receive SLNC-coded packets. Unselfish nodes increase their credits when they re-broadcast innovative content pieces to neighbors. Oppositely, selfish nodes decrease their credits as they overhear but reject to re-broadcast SLNC-coded content pieces. According to simulation results, SC stimulates these unselfish nodes for their generous content sharing but punishes selfish ones because they will run out of their credits eventually. To the best of our knowledge, SC is the first research that works on the integration of the credit-based incentive protocol and the SLNC-based cooperative content distribution.

The remainder of this paper is organized as follows. Section 2 discusses related work. Section 3 explains how to classify vehicular nodes into four types and their corresponding guidelines for stimulating incentives in SocialCode. Section 4 describes designs and flows of SocialCode in detail. Section 5 presents simulation results of SocialCode and well-known SLNC-based CCD approaches to show excellent performance of SocialCode. At last, Section 6 concludes this paper and discusses future research issues.

2. Related Work

In this section, well-known credit-based incentive systems are discussed later. The fairness principle for the credit-based incentive protocol is the major concern in Reference [23]. It presents a fair incentive protocol for four selfish conditions in MANETs. A trusted credit clearance service (TCCS) is used to perform trusted fair credit clearance when requested. As the destination has received the packet, TCCS charges credits values $(n - 1) \alpha_c + \beta_c$ from the source node. Then it will deposit credit value α_c and β_c to the first $(n - 1)$ intermediate nodes and the last one, respectively, where $\alpha_c < \beta_c$ because the last intermediate node must notify TCCS for clearing credits. In Reference [24], a mechanism is proposed to prevent selfish behavior according to the barter rules. It indicates that nodes may have different interests on different contents. There are two types of messages. The primary message is the one that a given node has interests in its content but the secondary one is a message that is not interesting to the given node. When two nodes enter communication range of each other, they exchange their message lists to check which messages they want to download from each other. This paper shows that selfishly downloading only those primary messages for a given node is not helpful for it later. It might be better for the node to download and carry the secondary messages such that the node can exchange them for the primary ones with other nodes. Obviously, these aforementioned methods will crash if the central control system fails.

With the receipt counting method (RCM) [25], if an intermediate vehicle receives a new packet forwarded from a preceding vehicle, it returns a receipt for this packet to the preceding one. As soon as the destination has received this packet, every intermediate forwarding node would demand overall payments with the source node, based on the number of all receipts. Thus, the source node has to pay each intermediate node the total claimed rewards through the central control system. Nonetheless, RCM undergoes the overspending problem, which means the source node is not able to limit its overall payments since it cannot estimate the number of total forwarding nodes ahead. It also suffers failure of the central control system.

The authors in Reference [26] propose the Fair Reimbursement and Motivating sweepstake scheme (FRAME) with a sweepstake component and a weighted rewarding one. The weighted rewarding component computes the weight and corresponding rewards of every vehicle. Besides, the sweepstake component awards the winning vehicle in the forwarding process a constant amount of reward. Nonetheless, nodes may refuse to forward packets owing to the insignificant rewards paid in FRAME. MobiCent [27] proposes a credit-based incentive system for the Disruption Tolerant Network (DTN) and computes payment by a multiplicative decreasing reward (MDR) scheme. It helps clients to minimize cost or minimize delay. The edge insertion and hiding attacks among nodes are solved by integrating both credit and cryptographic techniques. However, both the source and destination have to access the network for sending control messages all the time. Moreover, it needs a convincing third party like TCCS to handle the confirmation and payment functions. The reciprocal incentive scheme (RIS) [28] analyzes how two types of selfish nodes choose appropriate data objects to maximize their profits under buffer constraints. It also studies the relationship between object selections of the nodes and the stored network information. Each node in the opportunistic network is assumed selfish. A-type nodes can freely download various data objects from Internet but the ISP charges B-type ones with high downloading fee. With RIS, B-type nodes can request data objects from A-type ones with lower cost, instead of downloading them directly from the Internet. However, these credit-based incentive approaches do not work for the SLNC-coded CCD process.

Table 1 compares important characteristics of related credit-based incentive approaches with the proposed SocialCode. Except the Barter-based approach, SocialCode is the only one that considers different levels of node selfishness and interest and then classifies all nodes into four types. It also proposes corresponding distributed credit payment processes for these four types of nodes, instead of the centralized process adopted by most approaches. It further avoids the credit overspending problem of RCM and the reluctance to forward packets with FRMAE. Most importantly, SC strongly integrates its credit-based incentive mechanisms with the SLNC-based CCD to stimulate efficient content distribution in VANET, which is a significant advance in this field. On the other hand, MobiCent and RIS focuses on avoiding two kinds of security attacks and investigating how the scope of network information influences node decisions, respectively. Due to space limitation, SocialCode focuses on how to improve the conventional SLNC-based cooperative content distribution protocol and combine the proposed credit-based incentive approach to encourage all classes of vehicular nodes to rise their incentives for sharing content data. Consequently, security and privacy issues of the credit-based system are not discussed in this paper.

Table 1. Characteristics of related credit-based incentive approaches.

	Barter-Based	RCM	FRAME	MobiCent	RIS	SocialCode
Selfishness considered	Y	Y	Y	Y	Y	Y
Interest considered	Y	N	N	N	N	Y
Distributed credit control	N	N	N	N	Y	Y
Credit overspent	N	Y	N	N	N	N
Nodes may reluctant to forward packets	N	N	Y	N	N	N
Attacks considered	N	N	N	Y	N	N
Integrating with the SLNC-based CCD approach	N	N	N	N	N	Y

3. Concepts for Stimulating Incentives in SocialCode

In this section, we first discuss how to classify vehicular nodes into four types depending on their distinct levels of selfishness and interests on various data contents. Then we propose important guidelines for these four types of nodes in SocialCode to stimulate their incentives on forwarding content pieces in the CCD process.

3.1. Classification of Four Types of Vehicular Nodes in SocialCode

In real life, every node has distinct levels of selfishness and interests on various data contents. If the driver or any passenger of a vehicle in a given AoI needs the electronic coupon broadcast by the RSU, this vehicular node has a high interest on this coupon. Otherwise, its interest level on this coupon is low. If the vehicular node intends to join the CCD process by forwarding the received coupon to its neighbors, its selfishness level on this coupon is low. On the other hand, it has high selfishness on this coupon. Therefore, SocialCode categorizes vehicular nodes into four types, depending on combinations of their different interest and selfishness levels on specific data content. Corresponding operations for four types of vehicular nodes are listed in Table 2.

Table 2. Definitions and operations for four types of vehicular nodes.

	High Selfishness	Low Selfishness
High Interest	<i>Overhearing Node:</i> This kind of node stores received SLNC pieces and attempts to decode them. It does not forward them to its neighbors.	<i>Destination Node:</i> This kind of node stores received SLNC pieces and attempts to decode them. It re-encodes them and forwards re-encoded pieces to neighbors.
Low Interest	<i>Irrelevant Node:</i> This kind of node drops received SLNC pieces. It does not forward them to neighbors.	<i>Intermediate Node:</i> This kind of node stores received SLNC pieces. It just forwards them to neighbors.

3.2. SocialCode Guidelines for Distributed Incentive Stimulation

Whenever these four types of nodes have broadcast or received SLNC-coded content packets, the following rules of SocialCode must be considered first.

- (1) Only the node that has enough credit value can broadcast a SLNC-coded packet to its neighbor nodes.
- (2) Only the node that has enough credit value can receive a SLNC-coded packet from its neighbor nodes.
- (3) The receiving node must pay certain credits to encourage the sending one when it receives valuable content pieces.
- (4) The sending node must compensate the receiving one with certain credits if it broadcasts valueless content pieces.

Therefore, for satisfying the aforementioned rules to stimulate incentives of vehicular nodes on their CCD processes, SocialCode proposes the following guidelines, which are listed in Table 3 and clarified below, to modify credit values of these nodes.

- (1) As soon as the overhearing or destination node receives an innovative piece broadcast from its nearby intermediate or destination one, it will cache this innovative content piece in its output queue and issue an ACK packet to the sending node for paying its credit value α to increase that of the sending node, if its credit value is not less than α . Thus, Table 3 expresses this guideline as Innovative piece: $(+\alpha, -\alpha)$, which means SocialCode stimulates the sending node for sending an innovative piece. Otherwise, if the credit value of the destination or overhearing node is less than α , it cannot accept any innovative content piece from its neighbors.

- (2) Whenever the destination node receives a non-innovative piece broadcast from its nearby intermediate or destination one, it will drop this piece first. Then it will increase its credit value by β and issue an ACK packet to the sending node for requesting the credit value β from the sending node. Table 3 expresses this guideline as Non-innovative piece: $(-\beta, +\beta)$, which means SocialCode punishes the sending node for sending a non-innovative piece. Thus, if the credit value of the destination or intermediate node is less than β , it cannot re-broadcast any non-innovative piece to its neighbors. Oppositely, when the overhearing node receives a non-innovative piece, it will not issue an ACK packet to request any credit from the sending node. Table 3 expresses this guideline as Non-innovative piece: $(0, 0)$, which means SocialCode does not need to punish the sending node when it sends a non-innovative piece to the overhearing node, due to its selfish operation.
- (3) While the intermediate node finds the content piece it received already in its output queue, it will abandon this duplicate piece. Then it will increase its credit value by δ and issue an ACK packet to the sending node for requesting the credit value δ from the sending node. Therefore, Table 3 expresses this guideline as Duplicate piece: $(-\delta, +\delta)$, which means SocialCode punishes the sending node for sending a duplicate piece to the intermediate node. Otherwise, it will issue an ACK packet to the sending node for paying its credit value δ to increase that of the sending node. Thus, Table 3 expresses this guideline as Non-duplicate piece: $(+\delta, -\delta)$, which means SocialCode stimulates the sending node for sending a non-duplicate piece to the intermediate node. If the credit value of the intermediate node is less than δ , it cannot send any content piece, which may be a duplicate one, to its neighbors or accept any non-duplicate content piece from its neighbors.
- (4) Because the irrelevant node does not join the SLNC-based CCD process, it just drops the received piece and keeps its credit value unchanged. Hence, Table 3 expresses this guideline as Any piece: $(0,0)$.

Table 3. Guidelines to modify credit values and stimulate incentives for four types of nodes.

Receiver	Sender		
Destination Node	Intermediate Node	Overhearing Node	
Destination Node	Innovative piece: $(+\alpha, -\alpha)$ Non-innovative piece: $(-\beta, +\beta)$	Duplicate piece: $(-\delta, +\delta)$ Non-duplicate piece: $(+\delta, -\delta)$	Innovative piece: $(+\alpha, -\alpha)$ Non-innovative piece: $(0, 0)$
Intermediate Node	Innovative piece: $(+\alpha, -\alpha)$ Non-innovative piece: $(-\beta, +\beta)$	Duplicate piece: $(-\delta, +\delta)$ Non-duplicate piece: $(+\delta, -\delta)$	Innovative piece: $(+\alpha, -\alpha)$ Non-innovative piece: $(0, 0)$

Please note the value of α is larger than those of β and δ , that is, $\alpha > \beta > \delta$, which means destination nodes would earn more credits, that is, α , when they broadcast innovative SLNC-coded content pieces to nearby nodes than those, that is, β , paid when they broadcast non-innovative pieces. In this way, SocialCode stimulates destination nodes with higher credit values to encourage them to broadcast more innovative content pieces in the SLNC-based cooperative content distribution process, which raises the probability of the receiver to successfully decode all SLNC-coded content pieces and reconstruct original content data. However, SocialCode encourages intermediate nodes with lower credit values, that is, δ , to forward non-duplicate content pieces.

For the example shown in Figure 1, node A and C are destination nodes but node B and D are the overhearing and intermediate one, respectively. Assume the initial credit values of all these nodes are γ . Assume node C broadcasts a SLNC-coded content piece to its nearby nodes A, B and D at step 1. If this content piece is non-innovative to destination node A, A must reply an ACK packet with the credit value $-\beta$ to punish node C at step 2. After that, node A increases its credit value by β , which is subtracted from that of C. Overhearing node B has to send back an ACK packet with the credit value $+\alpha$ to stimulate node C, because it receives an innovative content piece. Similarly, when intermediate node D receives a non-duplicate piece, it will stimulate node C by sending it an ACK packet with the credit value $+\delta$. Hence, the total revenue of node C is $\alpha + \delta - \beta$. As node C receives all ACK packets

after broadcasting this content piece, credit values of node A, B, C and D are $\gamma + \beta$, $\gamma - \alpha$, $\gamma + \alpha + \delta - \beta$ and $\gamma - \delta$, respectively.

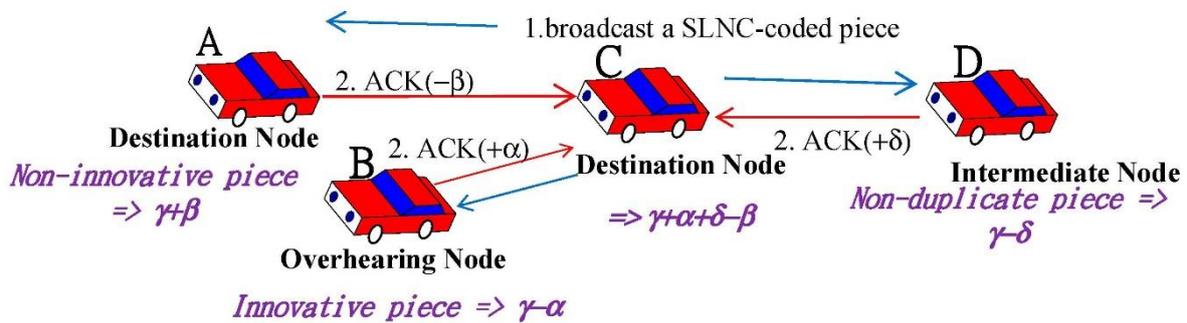


Figure 1. Credit values of node A, B, C and D after node C broadcasts a SLNC-coded content piece.

4. Design of SocialCode

In this section, we first describe the flow of SLNC-based cooperative content distribution in SocialCode. Second, we propose the content bitmap to identify differences on received content pieces of each destination node and use the *CalUtility* function to calculate their backoff times. Third, we explain how to use the dynamic forwarding priority assignment for fair scheduling among intermediate nodes. Finally, we illustrate the flow of SocialCode, including details of the proposed CCD broadcast and reception procedures.

4.1. SLNC-Based Cooperative Content Distribution of SocialCode

Social Code is based on the linear SLNC scheme, which operates as follows [18]. Whenever the sender node wants to distribute a multimedia content, it first encodes the content by splitting it into several generations, which includes g symbols. Then SLNC considers an original symbol as a vector over a base field and lets the sender apply a linear transformation on g symbols in a given generation to form a SLNC-coded packet, which is finally broadcast to nearby nodes. After receiving at least g linearly independent packets, the receiver can solve them to reconstruct original symbols. As the vehicular node enters the area covered by the RSU wireless signal, it accepts the SLNC-coded packets from RSU alone. Whenever it exits this area, it begins the SLNC-based CCD operations with nearby vehicles to share their received SLNC-coded packets.

There are two types of wireless channels, that is, the control channel (CCH) and the service channel (SCH), in VANET. SocialCode proposes three kinds of messages, that is, the beacon, the ACK and SLNC-coded one, for its SLNC-based CCD process. Figure 2 illustrates the flow of SocialCode, which alternates the control time slot with the service one.

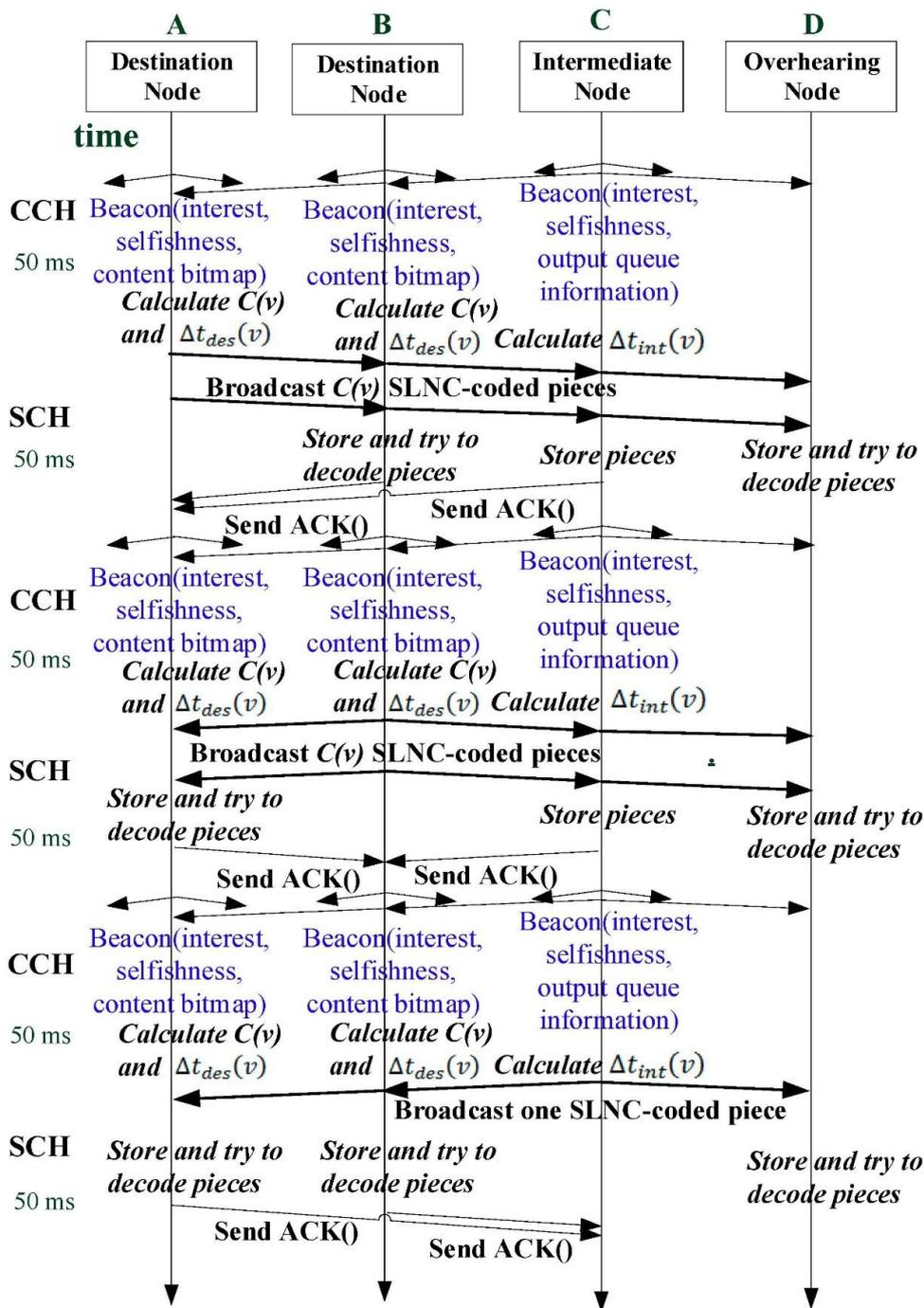


Figure 2. SLNC-based cooperative content distribution flow of SocialCode.

- (1) In every 50 ms time slot of CCH, all destination and intermediate nodes will broadcast their beacon messages to neighbors such that they will know numbers of destination and intermediate nodes in their proximities. All beacon messages contain the sender’s ID and selfishness/interest levels for distinct contents. SocialCode adopts in the content bitmap, which will be mentioned in Section 4.2, to record those SLNC content pieces received by the destination node during a preceding CCH time slot. In this way, the destination node, for example, nodes A and B in Figure 2, attaches its content bitmap in the beacon message to notify its neighbors of this information. Oppositely, the intermediate node includes its output queue information in its beacon message such that the intermediate node, for example, node C in Figure 2, could compute its dynamic forwarding priority, which will be mentioned in Section 4.3. However, the overhearing and

irrelevant nodes, for example, overhearing node D in Figure 2, will not broadcast any beacon message for this content data, because they do not forward receive SLNC pieces to neighbors.

- (2) After the time slot of CCH has expired, SCH starts its 50 ms service time slot. Among four types of vehicular nodes in SocialCode, only the destination node computes all three parameters, that is, the utility, the number of re-encoded content pieces and the backoff delay. As defined in Section 4.2, $C(v)$ represents the number of re-encoded content pieces for the destination node to forward during a service time slot. Further, $\Delta t_{des}(v)$ denotes the backoff delay of the destination node before it decides when to access channel. This value depends to the utility and the reception status, which is aggregated from the content bitmap, of the destination node. Whenever the backoff delay $\Delta t_{des}(v)$ of the destination node has expired, it, for example, node A, will broadcast $C(v)$ re-encoded content pieces as it senses an idle channel. Similarly, the intermediate node calculates its backoff delay $\Delta t_{int}(v)$, which will be discussed in Section 4.3. Since SocialCode gives the intermediate node the larger backoff delay, that is, $\Delta t_{int}(v)$, than $\Delta t_{des}(v)$, the intermediate node, for example, node C, can broadcast the content pieces it received to its neighbors only after every destination node has completed broadcasting its re-encoded content pieces, as shown in Figure 2. Please note that all destination and overhearing nodes will store received SLNC pieces and attempts to decode them due to their high interests on these pieces. Oppositely, the intermediate node only stores received SLNC pieces and broadcasts them one at a time in each successive service time slot.
- (3) After step 2, all destination and intermediate nodes will send ACK messages back to the sending nodes for modifying their credit values and stimulating their incentives to join the SLNC-coded CCD process of SocialCode, according to guidelines in Section 3.2.

Section 4.2 will explain details of how to calculate the utility, the number of re-encoded content pieces and the backoff delay for the destination and intermediate nodes

4.2. SocialCode Content Bitmaps among Destination Nodes

At every service time slot, each RSU repeatedly generates and broadcasts SLNC-coded content pieces to all nodes that can receive its wireless signals. SocialCode proposes to adopt one bit in the content bitmap to indicate whether a node has received a specific SLNC-coded piece broadcast by the RSU at a service time slot. In Figure 3, if nodes v and j have received SLNC-coded content pieces from time slot 142 to 146 and 144 to 149, respectively, the content bitmap, that is, BT^v , of node v from timeslot 142 to 149 is specified as [1 1 1 1 0 0] and that, that is, BT^j , of node j as [0 0 1 0 1 1 1]. Here a bit 1 indicates that node v or j has received the content piece issued at the corresponding time slot. For calculating the utility value of node v over node j , SocialCode invokes the proposed $CalUtility(v, j)$ function in Figure 4, where BT_i^v denotes the associated bit value of time slot i in the content bitmap of node v . Moreover, t_{max}^v and t_{min}^v indicate the starting and ending time slots in node v 's content bitmap, respectively. Hence, $CalUtility(v, j)$ calculates the number of bits with value 1 in the difference bitmap, that is, [1 1 0 1 0 -1 -1 -1] as shown in Figure 3, of node v over j . It finally returns 3 as the utility value of node v over j to express the number of innovative content pieces that node v could forward to j in the CCD process.

Timeslot	v	j	-	=	result
142	1	0			1
143	1	0			1
144	1	1			0
145	1	0			1
146	1	1			0
147	0	1			-1
148	0	1			-1
149	0	1			-1

Figure 3. The difference bitmap between content bitmaps of node v and j .

```

int CalUtility(v,j)
{
    Utility=0;
    tmin = min(tminv, tminj);
    tmax = max(tmaxv, tmaxj);
    for ( i=tmin; i ≤ tmax; i++) {
        if (( BTiv - BTij ) = 1)
            Utility++;
    }
    return Utility;
}

```

Figure 4. The CalUtility procedure for computing Utility.

Besides calculating the utility value over each nearby destination node using *CalUtility()*, destination node *v* further computes the aggregate utility, that is, *util(v)*, by summing all utility values of its nearby destination nodes by Equation (1), where *N(v)* contains every nearby destination node of *v*. After that, destination node *v* sorts aggregate utility values of all nearby destination nodes to arrange sequences for broadcasting their SLNC-coded content pieces. To allow the destination node with the highest aggregate utility value first accessing the wireless service channel, we propose Equation (2), which is modified from CodeOn [18], to let the backoff time, that is, $\Delta t_{des}()$, given at the start of every service time slot and the aggregate utility in inverse proportion. Here we define *N(v)* as the set of all nearby destination nodes of *v*. In Equation (2), three parameters, that is, $|N(v)|$, *K* and Δt_{max} , indicate the total number of nearby destination nodes of *v*, the total number of SLNC-coded pieces in a generation and the highest acceptable backoff time, respectively. At last, a random jitter generated by the function *Rand(0, T_j)* is summed up to Equation (2) to break the tie due to the same parameters. As soon as the backoff time of a destination node has expired, it broadcasts its re-encoded content pieces in a batch to all nearby nodes. Equation (3) computes the total number of re-encoded content pieces, that is, *C(v)*, by applying the ceiling function after dividing the aggregate utility, that is, *util(v)*, by the total number of nearby destination nodes, that is, $|N(v)|$. According to guidelines above, the destination or intermediate node must check whether its credit value is not less than $(|N(v)| \times \beta)$ before it re-broadcasts a SLNC-coded content piece, because it has to decrease its credit value by $(|N(v)| \times \beta)$ if this content piece is non-innovative to all $|N(v)|$ nearby nodes at the worst case. Otherwise, it must defer its transmission until next transmission opportunity.

$$util(v) = \sum_{j \in N(v)} CalUtility(v,j) \tag{1}$$

$$\Delta t_{des}(v) = \left[1 - \frac{util(v)}{K \times |N(v)|} \right] \times \Delta t_{max} + Rand(0, T_j) \tag{2}$$

$$C(v) = \frac{util(v)}{|N(v)|} \tag{3}$$

In summary, with the proposed content bitmap and the *CalUtility* function, SocialCode identifies differences on received content pieces of each destination node to calculate all utility values and the corresponding aggregate one, which means SocialCode can recognize the missing innovative content pieces the nearby destination node needs to support effective SLNC-based CCD. On the other hand, each node in traditional schemes [18–21] does not distinguish content pieces it has received from those of nearby nodes. Instead, it adopts the difference on the number of received content pieces between it and nearby nodes to calculate utility values.

4.3. The Dynamic Forwarding Priority Assignment for Fair Scheduling among Intermediate Nodes

According to the incentive stimulation guideline proposed in previous subsection, there are two cases that the credit value of the sending intermediate node will decrease. First, if the sending intermediate node delivers a non-innovative piece to the receiving destination one, it must pay its credit value + β to the receiving node. Second, the sending intermediate node must pay its credit value + δ to the receiving intermediate one if it delivers a duplicate piece to the receiving one. Hence, if an intermediate node keeps delivering like this and undergoes such the credit loss, it will dissipate its credit values soon. For alleviating this defect, SocialCode must assign dynamic priorities to all nearby intermediate nodes by Equation (4) and propose a fair scheduling method for them to deliver content piece by turns. SocialCode adopts two important parameters of intermediate node i at time t to decide the forwarding priority, that is, pri_i^t . They are the number of cached content pieces, that is, the queue length and the current *time-to-live* (TTL) value of the first content piece stored in its output queue, which are denoted by q_i^t and tll_i^t , respectively. Please note that ω ($0 \leq \omega \leq 1$) is the factor to give the weight to the queue length, q_{max} is the maximum number of content pieces that can be stored in the output queue and tll_{max} is the initial TTL value of every content piece.

$$pri_i^t = \omega \times \frac{q_i^t}{q_{max}} + (1 - \omega) \times \frac{(tll_{max} - tll_i^t)}{tll_{max}} \quad (4)$$

As mentioned above, SocialCode should assign a longer backoff delay to the intermediate node than that of each destination node. Hence, the intermediate node could deliver content pieces stored in it after all destination nodes have completed broadcasting their re-encoded NC content pieces, as shown in Figure 1. SocialCode further proposes Equation (5) to assign the backoff delay, that is, $\Delta t_{int}(i)$, to intermediate node i . For letting the intermediate node broadcast content pieces it received to its neighbors after every destination node does, the first part of $\Delta t_{int}(i)$, that is, $\Delta t_{max} + T_j$, must be larger than the backoff delay, that is, $\Delta t_{des}(v)$ in Equation (2), of each destination node v in $N(v)$. The second part of $\Delta t_{int}(i)$, that is, $\frac{1}{Pri_i^t} \times \Delta t_{max}$, is inverse proportional to its forwarding priority, that is, pri_i^t , calculated by Equation (4). Based on Equation (5), SocialCode assigns a smaller backoff time, that is, $\Delta t_{int}(i)$, to intermediate node i that has a larger forwarding priority, that is, pri_i^t . Consequently, the intermediate node with a larger forwarding priority will forward its stored content pieces earlier than other intermediate nodes with smaller priorities do

$$\begin{aligned} \Delta t_{int}(i) &= \left(1 + \frac{1}{Pri_i^t}\right) \times \Delta t_{max} + T_j \\ &= (\Delta t_{max} + T_j) + \left(\frac{1}{Pri_i^t} \times \Delta t_{max}\right) \\ &> \Delta t_{des}(v) + \left(\frac{1}{Pri_i^t} \times \Delta t_{max}\right), \text{ where } v \in N(v) \end{aligned} \quad (5)$$

4.4. The Flow of SocialCode

Figure 5 illustrates the main flow diagram of SocialCode, which consists of two kinds of operations depending on whether the vehicular node can receive the wireless signal of the RSU or not. Assume the maximum period for the vehicular node to stay in contact with the nearby RSU is t .

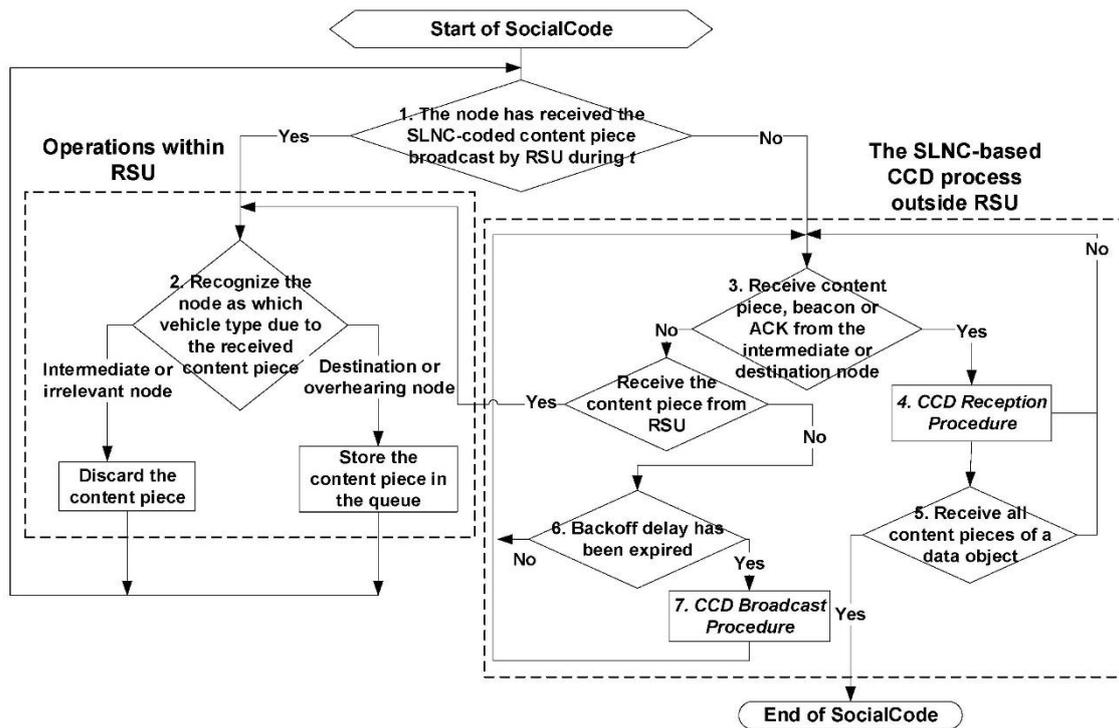


Figure 5. The main flow diagram of SocialCode.

- (1) Whenever a node has received a SLNC-coded content piece broadcast by the RSU during t , it executes the first kind of SocialCode operations as follows. It will use its interest and selfishness value on this received piece to recognize itself as which type of vehicular node according to rules of Table 2.
- (2) If it identifies itself as the intermediate or irrelevant node, it will discard this received piece. On the other hand, it will store this piece in its queue if it is the destination or overhearing node. Afterward, it continues receiving the content piece broadcast by the RSU and goes back to step 1.
- (3) If the node does not stay in the wireless signal scope of any RSU at step 1, which means it has not received any content piece broadcast by any RSU for the period of t , it will start the second kind of SocialCode operations to perform the SLNC-based CCD process between it and nodes also losing contacts with RSUs.
- (4) As soon as the node receives the SLNC-coded content piece, the beacon message or the ACK packet from any nearby intermediate or destination node during this process, it will invoke the CCD reception procedure to receive decode all SLNC-coded pieces of a generation.
- (5) If the node has received and decoded all content pieces of a data object, SocialCode has completed its flow. Otherwise, this node goes back to step 3 to repeat the SLNC-based CCD process.
- (6) Oppositely, if the node receives a fresh content piece broadcast by any RSU, it stops executing the SLNC-based CCD process and goes back to step 2, which recognizes itself as which type of vehicular node as described above.
- (7) If the node has not received a fresh content piece broadcast by an RSU and its backoff timer has been expired, this intermediate or destination node will invoke the CCD broadcast procedure to forward or re-broadcast content pieces stored in its queue. Afterward, this node goes back to step 3 to repeat the SLNC-based CCD process.

Details of the CCD broadcast and reception procedures will be explained as follows.

4.4.1. The CCD Broadcast Procedure

As shown in Figure 6, steps of the CCD broadcast procedure are listed below. Please note that intermediate node v forwards its first SLNC-coded piece in queue only after each destination node i has re-broadcast $C(i)$ pieces because its backoff timers, that is, $\Delta t_{int}(v)$, is longer than that, that is, $\Delta t_{des}(i)$, of each destination node i .

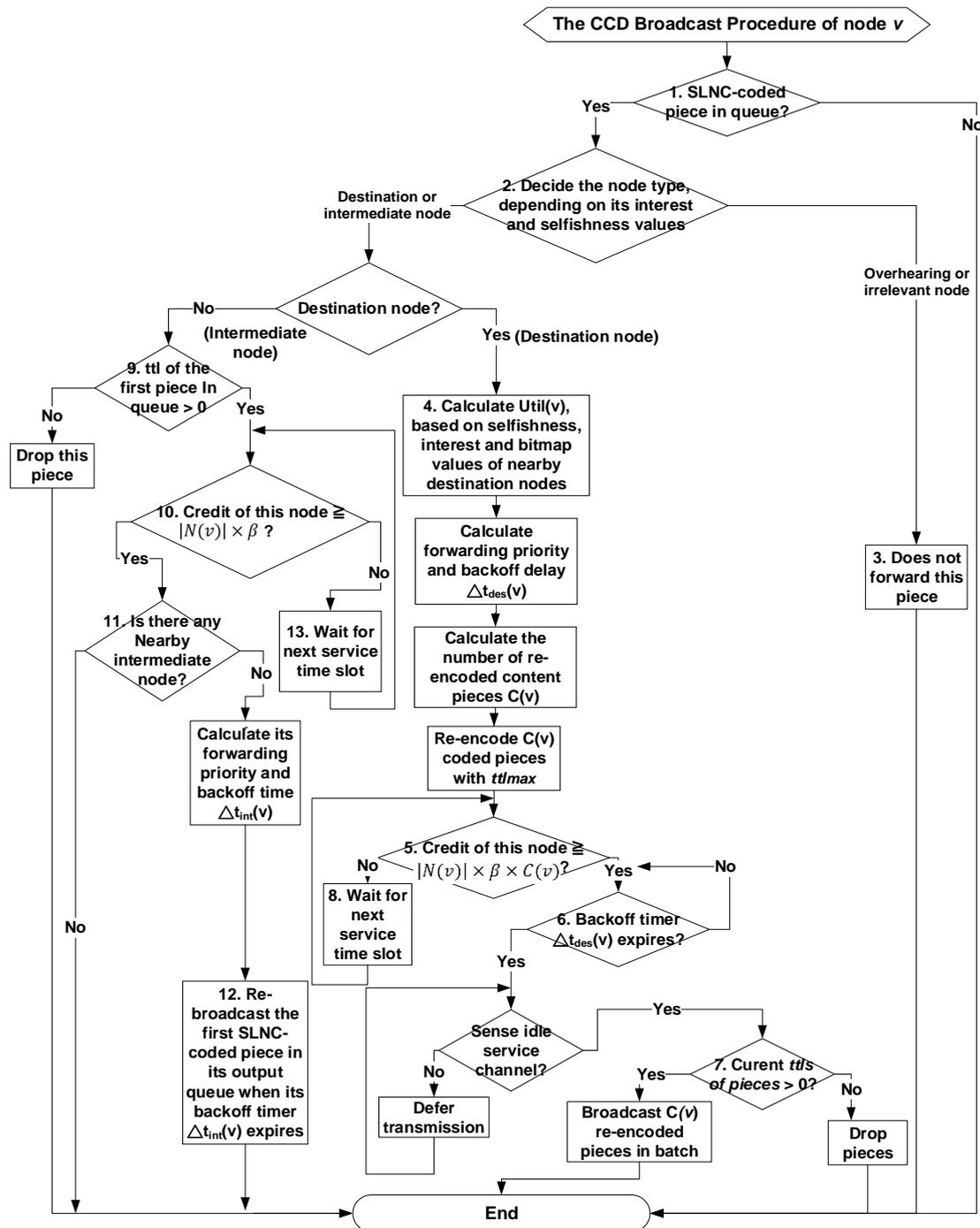


Figure 6. Flow of the CCD broadcast procedure.

- (1) When node v starts the CCD broadcast procedure, it will check whether its output queue has cached SLNC-coded pieces. If no, this node has completed its CCD broadcast procedure.

- (2) Otherwise, this node decides its node type, according to its selfishness and interest values on SLNC-coded content pieces stored in its output queue.
- (3) If it is the overhearing or irrelevant node, it will not re-broadcast the piece to its nearby nodes. This node then exits its CCD broadcast procedure
- (4) If this node is the destination node, it will first adopt Equation (1) to calculate its aggregate utility, that is, $util(v)$, based on the selfishness, interest and content bitmap values carried in beacon messages, which are issued by nearby destination nodes. Then, this node will use Equations (2) and (3) to calculate its backoff delay, that is, $\Delta t_{des}(v)$ and the number of re-encoded content pieces, that is, $C(v)$, respectively. Finally, it re-encodes $C(v)$ content pieces, which initial ttl values are set as ttl_{max} .
- (5) After that, this node has to check whether its credit value is not less than $|N(v)| \times \beta \times C(v)$, because it must re-broadcast all $C(v)$ pieces in one batch [18,19].
- (6) If yes, it then defers its transmission until its backoff timer expires. At this time, if this node senses an idle service channel, it will update current ttl values of all $C(v)$ re-encoded content pieces.
- (7) Later on, it re-broadcasts these content pieces only when their current ttl values are greater than 0. If ttl values of these content pieces are equal to 0, this node will drop these pieces. At this time, this node has finished its CCD broadcast procedure.
- (8) If the credit value of this node is less than $|N(v)| \times \beta \times C(v)$ at step 5, it has to wait for next service time slot and goes back to step 5.
- (9) If node v is the intermediate node after step 2, it will check whether the ttl value of this received piece is larger than 0. If no, it drops this piece and exits the CCD broadcast procedure.
- (10) If the ttl value of this received piece is larger than 0, this node has to check whether its credit value is less than $|N(v)| \times \beta$.
- (11) If the credit value of intermediate node v is not less than $|N(v)| \times \beta$ at step 10 and there is any nearby intermediate node, it adopts Equations (4) and (5) to compute its forwarding priority, that is, pri_v^t and backoff delay, that is, $\Delta t_{int}(v)$, respectively.
- (12) When the backoff timer, that is, $\Delta t_{int}(v)$, of intermediate node v expires, it will broadcast the first SLNC-coded content piece in its queue. Details operations of this step are similar to those of steps 6 and 7 for destination nodes. After that, this node has finished its CCD broadcast procedure.
- (13) If the credit value of this intermediate node is less than $|N(v)| \times \beta$ at step 10, it has to wait for next service time slot and goes back to step 10.

4.4.2. The CCD Reception Procedure

The flow of the CCD reception procedure is shown in Figure 7. Its details are described as follows.

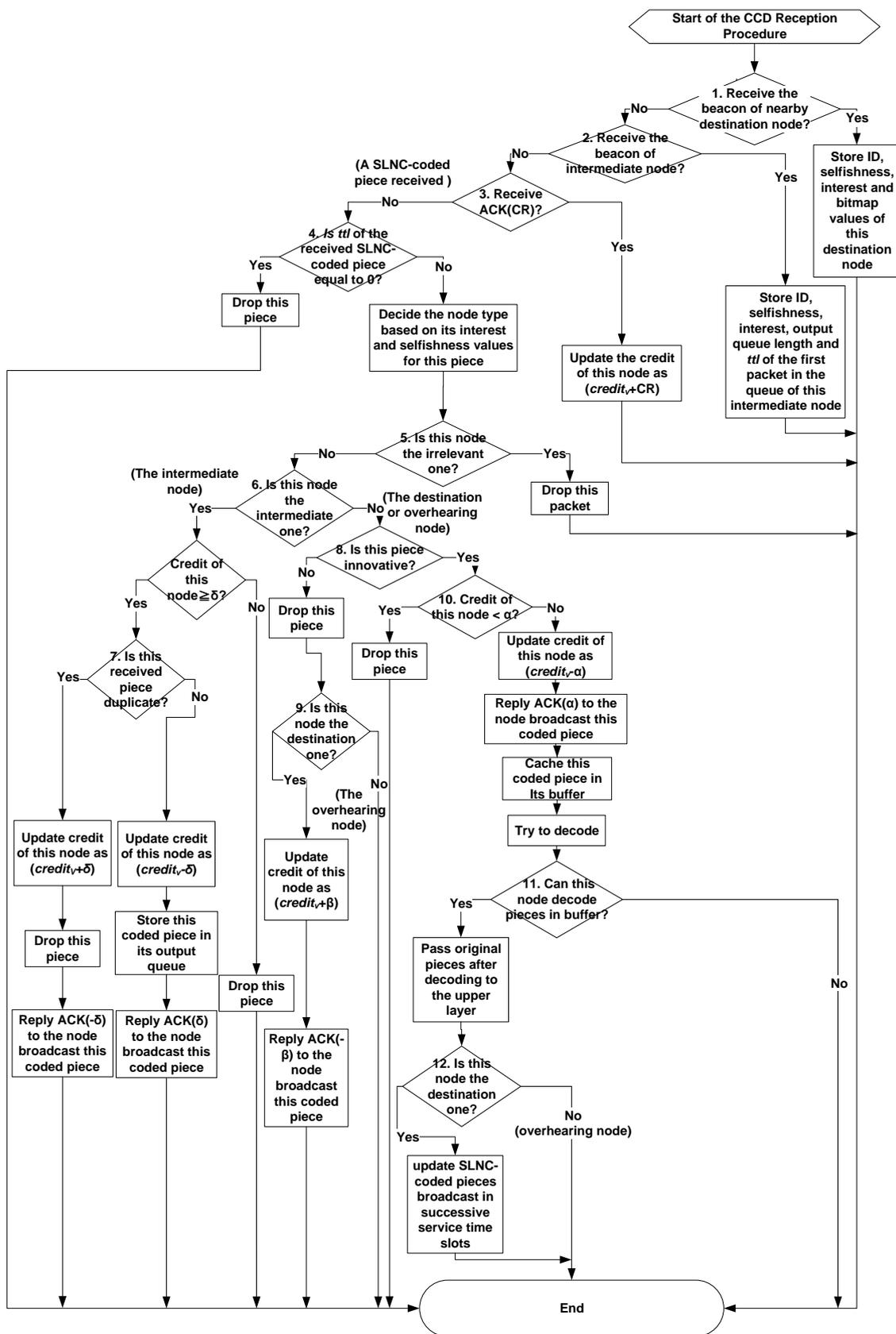


Figure 7. Flow of the CCD reception procedure.

- (1) When node v receives the beacon message sent by a destination node, it will store the ID, selfishness, interest and bitmap values carried in this beacon message. At this time, this procedure has been finished.
- (2) If this beacon message is issued by an intermediate node, node v stores the ID, selfishness, interest and queue length of this intermediate node and the ttl value of the first packet in the output queue of this sending node. Then node v exits this procedure.
- (3) If node v receives an ACK(CR) packet which carries the credit value CR, it will update its credit value $credit_v$ as $(credit_v + CR)$ and stop this procedure. For example, when the nearby destination node has successfully received an innovative SLNC-coded piece broadcast by node v in the preceding service time slot, it would subtract α from its credit value and send an ACK(α) back to node v for increasing the credit value of node v by α .
- (4) If node v does not receive an ACK(CR) packet at step 3, it further checks whether the ttl value of the received packet, that is, the SLNC-coded piece, is zero or not. If yes, it must drop this piece. Otherwise, it will decide its node type based on its interest and selfishness values for this piece.
- (5) If node v is the irrelevant node, it will drop this piece and exit this procedure.
- (6) If node v is the intermediate node, it will check whether its credit value is not less than δ . If no, it will drop this piece and exit this procedure.
- (7) If node v finds this piece is a duplicate one, it will add δ to its credit value, drop this piece and finally reply an ACK($-\delta$) to punish the nearby node that broadcasts this duplicate piece. Otherwise, it will subtract δ from its credit value, store this piece in its output queue and finally reply an ACK(δ) to stimulate the node that forwards this non-duplicate piece. Then node v exits this procedure.
- (8) If the type-checking operation at step 4 recognizes node v is not the irrelevant or intermediate one, this node must further examine whether the received SLNC-coded piece is innovative to it. If no, it will drop this piece.
- (9) After this, there are two cases to handle. First, if node v is a destination one, it will increase its credit value by β and reply an ACK($-\beta$) to punish the node broadcast this non-innovative piece. Second, if node v is an overhearing one, it will not issue an ACK packet to request any credit from the sending node when it receives a non-innovative piece, according to guideline 2 in Section 3.2. Then node v exits this procedure.
- (10) If node v receives an innovative SLNC-coded piece at step 8 and its credit value is less than α , it must drop this piece and exit this procedure, even this piece is innovative to it. Otherwise, it will subtract α from its credit value, reply an ACK(α) to stimulate the node broadcast this coded piece, cache this piece in its decoding buffer and try to decode buffered pieces.
- (11) If the original content pieces can be decoded from buffered pieces, node v will pass them to the upper layer. Otherwise, node v stops this procedure.
- (12) Further, if this node is the destination one, it will update the SLNC-coded pieces that will be broadcast in successive service time slots when it has recovered the original content pieces. If this node is the overhearing one, it will not re-broadcast the received pieces, due to its high selfishness on this content data. After this step, node v has finished its CCD reception procedure.

5. Simulations

5.1. Simulation Setup

As shown in Figure 8, we use the $10524 \times 6924 \text{ m}^2$ experiment area to perform our simulations in this paper. This area includes part of the urban map of Taipei, Taiwan and contains forty road segments and twenty-five junctions. Assume four RSUs at different junctions regularly broadcast content pieces that are encoded by SLNC from a file. This experiment adopts two simulation tools. The first one is the Simulation of Urban Mobility (SUMO) tool [37], which creates 216, 867 and 3468 randomly distributed vehicular nodes and their corresponding trajectories based on the car-following

model. The instant velocity and speed acceleration of every vehicle are uniformly distributed in the interval $(0, 15)$ m/s and $(-0.5, +0.5)$ m/s², respectively. Afterward, we use the popular network simulator, NS-2.34 [38], to execute simulations for the SLNC-based CCD procedures, according to simulation parameters listed in Table 4. Assume every vehicular node and RSU adopt IEEE 802.11p and the Nakagami physical propagation model [39] where the energy detection range (ER) the data communication range (CR) are 700 m and 250 m, respectively. For fair comparisons in this paper, we use the same SLNC parameter values as those in CodeOn [18]. First, we adopt 3 Mbps and 12 Mbps to be the base rate for broadcasting beacons in the control channel and the data rate for forwarding SLNC-coded content pieces in the service channel, respectively. Time slots of both control and service channels are 50 ms. Moreover, the number of content pieces in a SLNC generation, the initial TTL value, the size of a SLNC-coded content piece and the output queue length of each vehicular node are 32 content pieces, 60 s, 1 Kbytes and 100 content pieces, respectively. Finally, Δt_{max} and T_j in Equation (2) are set as 2 ms and 100 μ s, respectively.

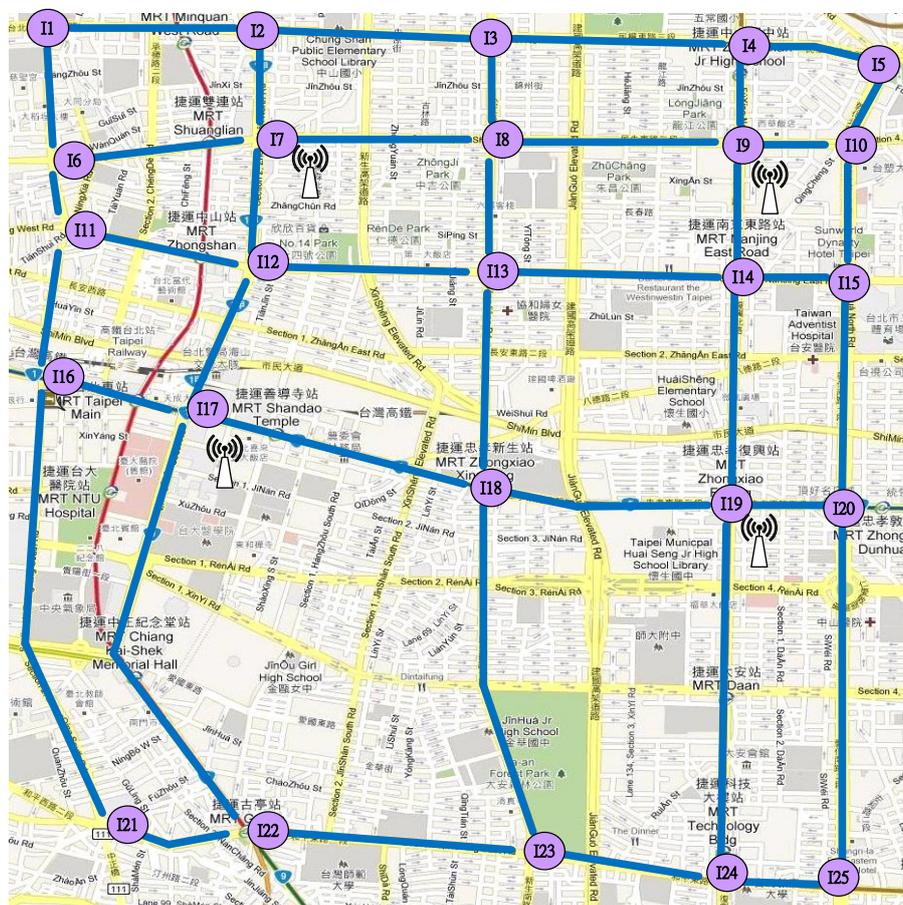


Figure 8. The experiment area used in simulations.

Table 4. Simulation parameters and their values.

Parameters	Value
Simulation Area	10524 × 6924 m ² area in urban Taipei, Taiwan, with 25 junctions and 40 road segments
MAC layer	802.11p
Propagation model	Nakagami m = 3
Data communication range (CR)	250 m
Energy detection range (ER)	700 m
Base rate in the control channel	3 Mbps
Data rate in the service channel	12 Mbps
The number of APs	4
Generation size	32 pieces
TTL of a SLNC-coded content piece	60 s
The size of a SLNC-coded content piece	1 KBytes
Output queue length	100 pieces
t_{max}, T_j	2 ms, 100 μs
Initial credit γ	50
$\alpha, \beta, \delta, \omega$	0.2, 0.04, 0.01, 0.5
File size	10, 6, 2 MBytes
The number of vehicular nodes	3468, 867, 216

In this paper, we will compare four schemes for evaluating their performance on stimulating content-sharing incentives for twenty runs as follows. The first two schemes are SocialCode (SC) and its variation, that is, SocialCode without Dynamic priority (SC without DP). The third and fourth ones, that is, RCM + CodeOn and FRAME + CodeOn, are two combinations of the traditional SLNC-based CodeOn and two credit-based incentive schemes, that is, RCM and FRAME, respectively. Please note that simulation results of RCM + CodeOn and FRAME + CodeOn in Figures 9–14 are denoted as RCM and FRAME respectively due to the limited space in these figures. Here, we compare three performance metrics of these four schemes with respect to three different file sizes, that is, 10, 6 and 2 M bytes (MB) and three different numbers of vehicle nodes, that is, 3468, 867 and 216, where 6 MB and 867 nodes are their default values in simulations. The probability for every vehicular node to become one of the intermediate, destination and overhearing nodes is equal. Each node has the same initial credit ($\gamma = 50$). Both RCM + CodeOn and FRAME + CodeOn set one unit of reward as 0.2. FRAME + CodeOn further fixes its probability to win the sweepstake and its extra sweepstake reward as 0.5 and 0.1, respectively. We also assign 0.2, 0.04, 0.01 and 0.5 to be the default value of α , β , δ and ω , respectively, as done in FRAME. Definitions of these three metrics are listed as follows.

- (1) *Average decoding percentage*: this metric represents the average of the percentages to decode SLNC-coded content pieces back to the original file for all destination nodes as the time elapses;
- (2) *Average file downloading delay*: this metric denotes the average of all elapsed time for the destination node to decode content pieces back to the original file completely;
- (3) *Average credits*: this metric expresses the average of all credits owned by the intermediate, overhearing and destination nodes as the time elapses.

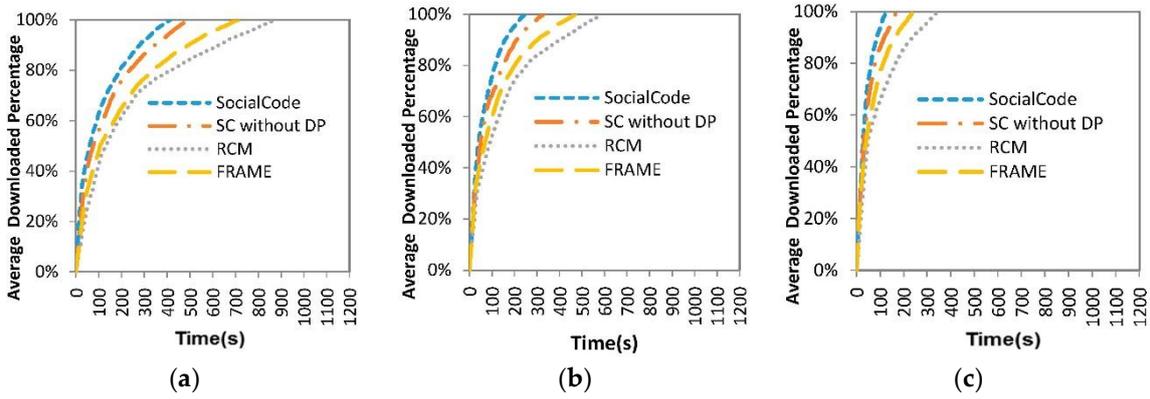


Figure 9. Average downloading percentage vs. file size (a) 10 MB; (b) 6 MB; (c) 2 MB.

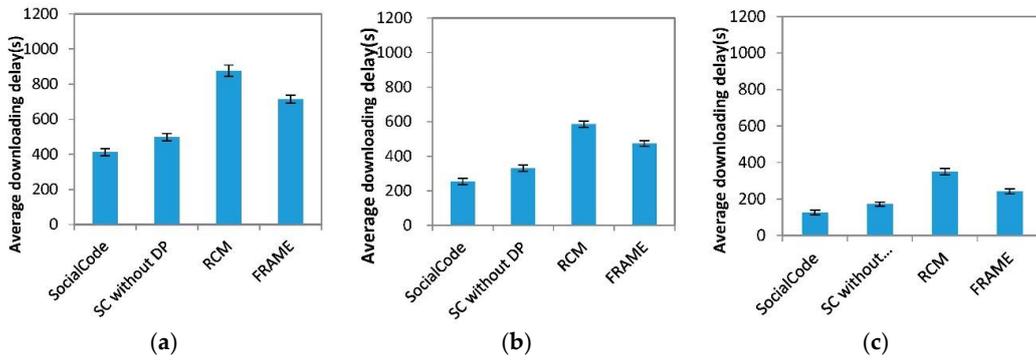


Figure 10. Average downloading delay vs. file size (a) 10 MB; (b) 6 MB; (c) 2 MB.

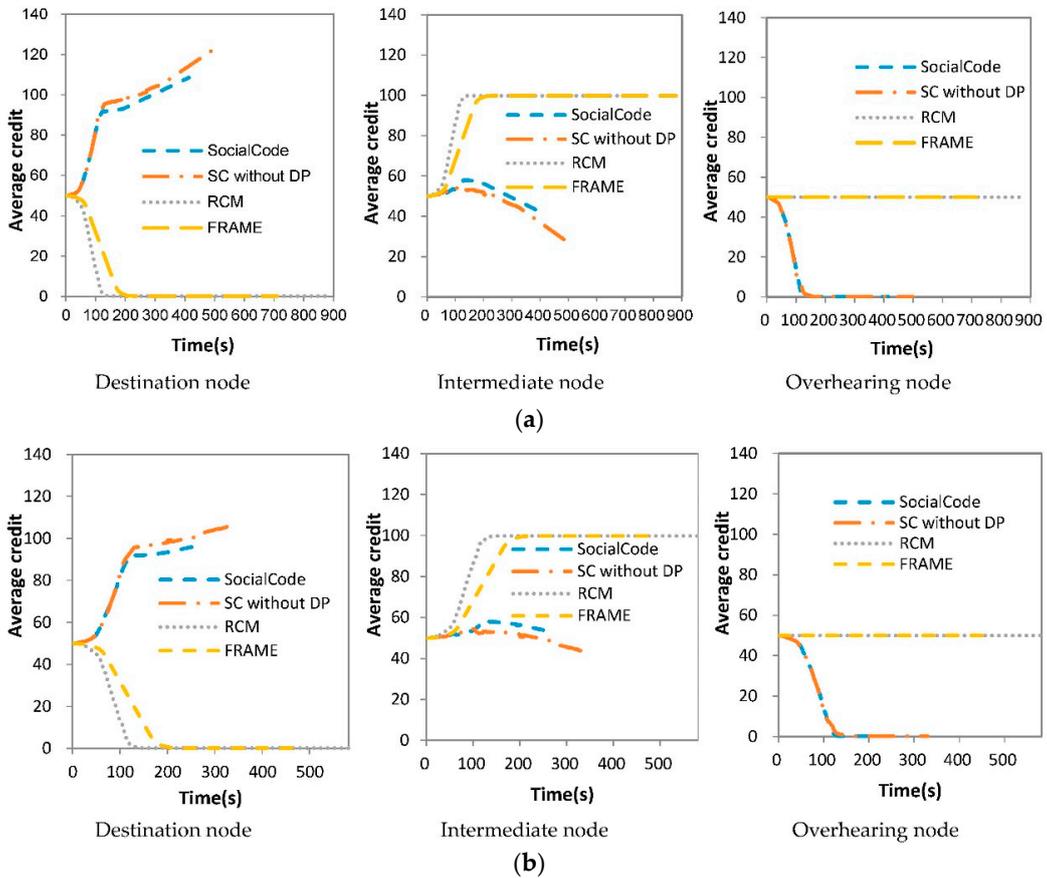


Figure 11. Cont.

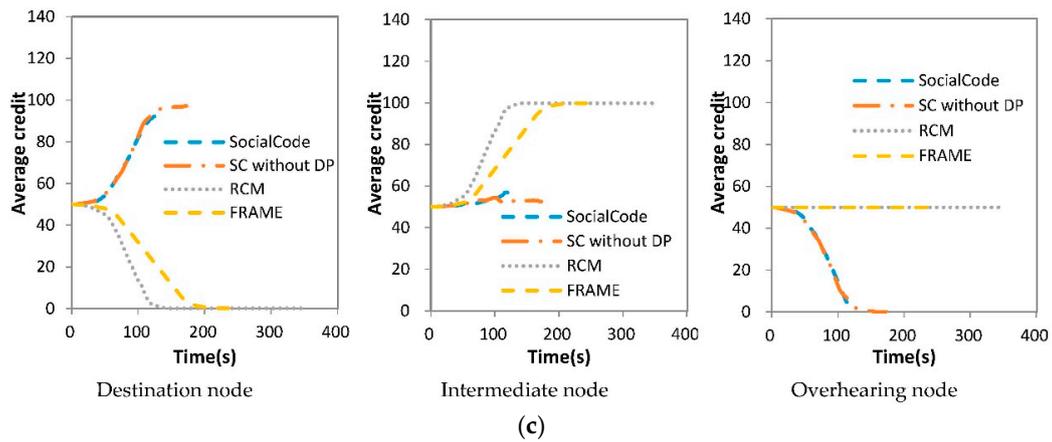


Figure 11. Average credit vs. file size (a) 10 MB; (b) 6 MB; (c) 2 MB.

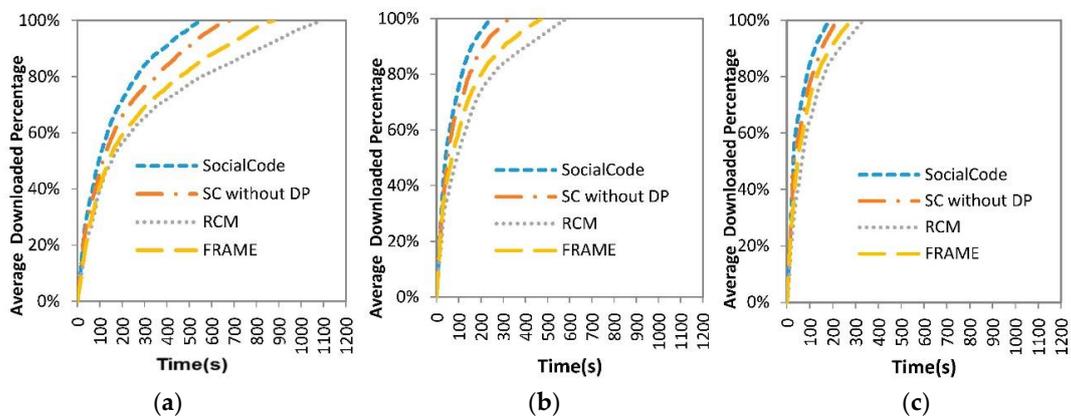


Figure 12. Average downloading percentage vs. the number of nodes (a) 3468 nodes; (b) 867 nodes; (c) 216 nodes.

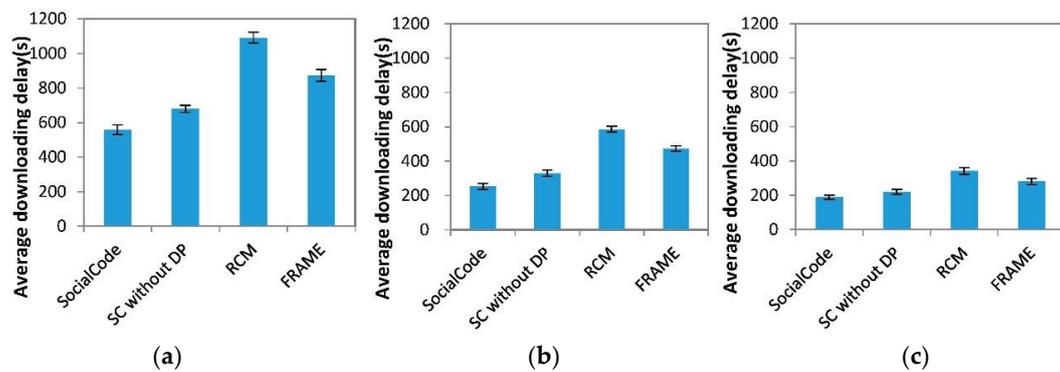


Figure 13. Average downloading delay vs. the number of nodes (a) 3468 nodes; (b) 867 nodes; (c) 216 nodes.

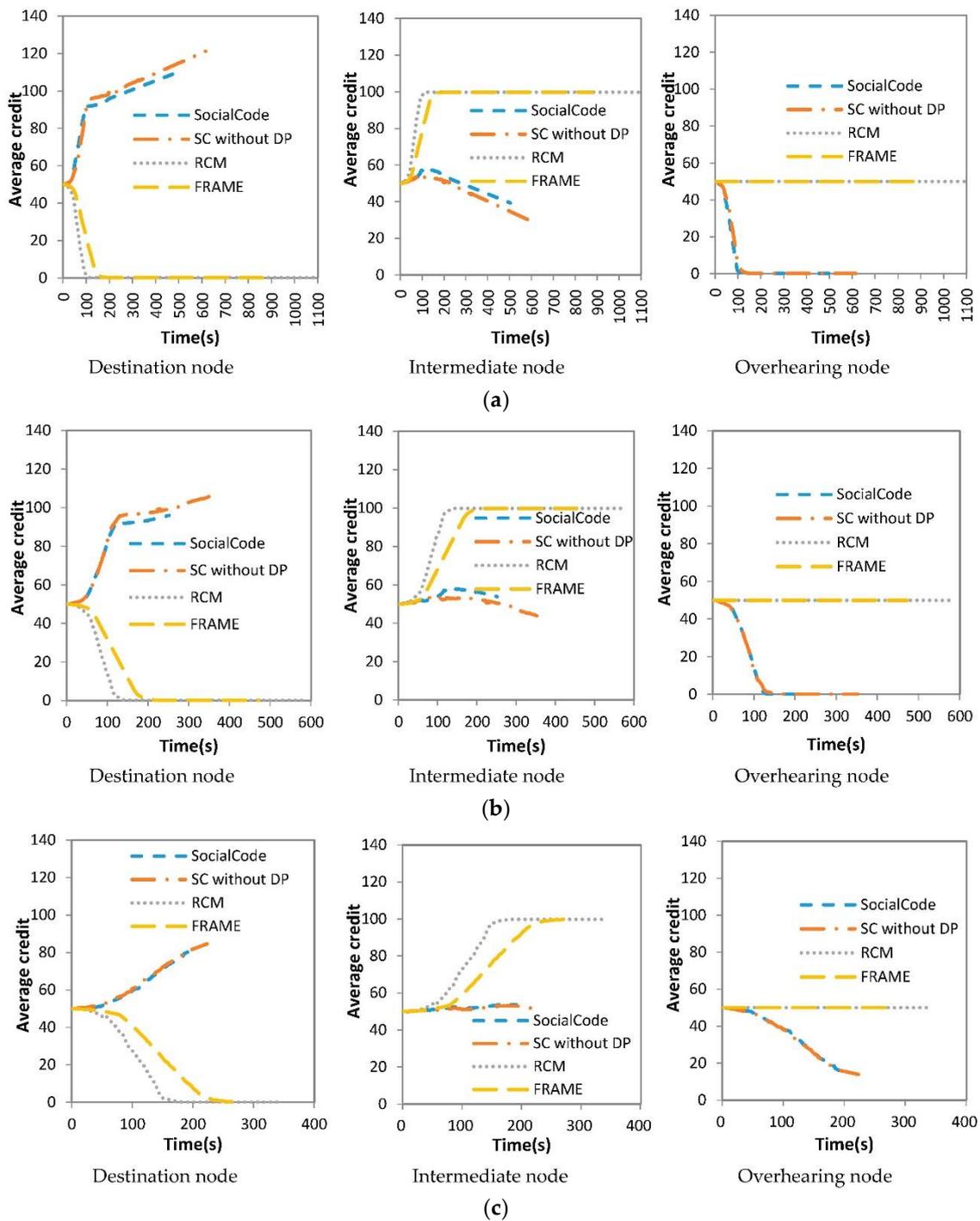


Figure 14. Average credit vs. the number of nodes (a) 3468 nodes; (b) 867 nodes; (c) 216 nodes.

5.2. Simulation Results for Different File Sizes

Here we compare three CCD metric values under three different file sizes, that is, 10 M, 6 M and 2 M bytes, for SocialCode, SC without DP, RCM + CodeOn (RCM for short) and FRAME + CodeOn (FRAME for short). It is obvious that the larger file has smaller decoding percentage and longer file downloading delay to decode enough SLNC-coded pieces back to the original file, as compared to those of the smaller ones, which are shown in Figures 10 and 11. For example, average file downloading delays of SC for 10 MB, 6 MB and 2 MB files are about 412, 253 and 125 s, respectively. Further, no matter the file size is, average decoding percentages of SC without DP, RCM + CodeOn and FRAME + CodeOn

are smaller than those of SocialCode, as shown in Figure 9. Further, their average file downloading delays are longer than those of SocialCode, which means the average file downloading delays of RCM + CodeOn > FRAME + CodeOn > SC without DP > SocialCode, as shown in Figure 10. As mentioned above, the vehicular node adopting both SCs first exchanges its content bitmap with all nearby nodes and then calculates the correct number of distinct SLNC-coded content pieces, the backoff time and the exact number of re-encoded pieces by the proposed CalUtility procedure and Equations (1)–(3). Hence, SocialCode and its variety achieve better performance than RCM + CodeOn and FRAME + CodeOn do on both metrics. Moreover, by assigning dynamic forwarding priorities with Equation (4) and backoff delays, that is, $\Delta t_{int}(i)$, with Equation (5), SocialCode could prevent a particular intermediate node forwarding its pieces for a long time, which in turn uses up its credits and stops it to forward any content piece at last. Consequently, SocialCode has more excellent decoding percentages and file downloading delays than SC without DP does. Oppositely, because both RCM + CodeOn and FRAME + CodeOn let the destination node pay its credits to the intermediate node for forwarding the SLNC-coded content pieces, some destination nodes may exhaust their credits soon, which terminates their cooperative content distribution processes. In this situation, they cannot broadcast and receive content pieces to and from nearby nodes when they are outside the signal range of the RSU. Hence, RCM + CodeOn and FRAME + CodeOn suffer from poorer decoding percentages and file downloading delays than SocialCode and SC without DP. Table 5 lists normalized ratios for average file downloading delays of SocialCode, SC without DP and FRAME + CodeOn over that of RCM + CodeOn under three file sizes. It shows that SocialCode achieves much lower normalized ratio for the file downloading delay than SC without DP, RCM + CodeOn and FRAME + CodeOn do when the file size is smaller.

Table 5. Normalized ratios for average file downloading delays of SocialCode, SC without DP and FRAME + CodeOn (FRAME) over that of RCM + CodeOn (RCM) under three file sizes.

Normalized Ratio of File Downloading Delay	File Size		
	10 MB	6 MB	2 MB
SocialCode	0.47	0.43	0.36
SC without DP	0.57	0.56	0.49
FRAME + CodeOn (FRAME)	0.81	0.81	0.69
RCM + CodeOn (RCM)	1.00	1.00	1.00

With respect to three different file sizes, average credits of three types of nodes, that is, the destination, intermediate and overhearing ones, for SocialCode, SC without DP, RCM + CodeOn and FRAME + CodeOn are illustrated in Figure 11a–c, respectively. The destination node with RCM + CodeOn needs compensating all intermediate nodes one unit of its credit for broadcasts a content piece to them. Thus, RCM + CodeOn consumes the credits of the destination node quicker than FRAME + CodeOn does, which is shown in Figure 11. Oppositely, FRAME + CodeOn has a slower rate to drain the credits of the destination node than RCM, because it will reward the winning vehicular node some definite credits by its sweepstake component. According to guidelines in Table 3, as the destination node with SocialCode and its variety forwards an innovative SLNC-coded piece to nearby destination or overhearing ones, it will earn credit α from them, which occurs frequently in the beginning of the CCD procedure. Hence, the average credits of the destination nodes raise faster during the period from time 50 to 150, as shown in Figure 11. As a result, SocialCode and SC without DP considerably extend lifetimes of the destination nodes for encouraging them to join the CCD procedure.

As mentioned above, when the intermediate node with RCM + CodeOn and FRAME + CodeOn receives a content piece from the destination node, it will increase its credits by one unit, which is paid by this destination one. As the intermediate node with SocialCode and its variety receives a non-duplicate content piece, it has to compensate the sending node credit δ due to the guideline in Table 3. Nevertheless, whenever it further forwards its content piece, which is non-duplicate to next intermediate node or innovative to next destination node, SocialCode and SC without DP will

raise its credit by δ or β , respectively, which is paid by next receiver accordingly. Hence, SocialCode and SC without DP achieve more steady values on average credits of the intermediate node than RCM + CodeOn and FRAME + CodeOn do, as shown in Figure 11. On the other hand, because SC without DP does not decide the dynamic priority of each intermediate node, the intermediate node that receives content pieces earlier has higher probability to forward its cached content pieces to nearby ones, which dissipate its credit values faster. Thus, as shown in (a–c) of Figure 11, the intermediate node of SC without DP suffers from lower and less stable average credits than that of SocialCode but its destination node owns higher average credits. Further, the larger the file is, the lower average credits the intermediate node owns but the higher ones the destination node does. Consequently, these results show that the dynamic forwarding priority assignment, proposed in Section 4.3, adopted by SocialCode achieves much fairer content piece forwarding among intermediate nodes than SC without DP does.

At last, as shown in Figure 11, RCM + CodeOn and FRAME + CodeOn keep average credits of their overhearing nodes to the value, that is, 50, of the initial credit γ because these nodes do not include in their CCD procedures. Based on the guideline in Table 3, the overhearing node with SocialCode and SC without DP will decrease its credit by α when it accepts an innovative piece from the destination or intermediate ones. Hence, its credit will decrease to zero rapidly, which is about 150 ms in Figure 11, since it does not earn any credit from nearby nodes by further re-broadcasting the received piece to them. As a result, SocialCode and SC without DP will prohibit the overhearing node from broadcasting its own pieces and receiving any successive piece. This means that SocialCode and SC without DP penalize the overhearing node due to its selfish operation during the CCD procedure.

5.3. Simulation Results for Different Numbers of Vehicular Nodes

Here we compare three metric values versus three numbers of vehicular nodes, that is, 3468, 867 and 216, to perform the CCD procedure using SocialCode, SC without DP, RCM + CodeOn and FRAME + CodeOn. In this simulation, the number of the destination nodes that distribute the file is proportional to that of total vehicular nodes. Hence, the larger the number of the destination nodes is, the higher the packet collision probability among nearby destination nodes is [18]. This in turn diminishes the average decoding percentage and raises the average file downloading delay of the destination node for all four schemes, as shown in Figures 12 and 13. For example, average downloading delays of SocialCode versus 3468, 867 and 216 nodes in Figure 13 are about 558, 253 and 188 s, respectively. Further, no matter the total number of nodes is, average decoding percentages of RCM + CodeOn, FRAME + CodeOn and SC without DP are smaller than those of SocialCode, as shown in Figure 12. Oppositely, their average downloading delays are larger than those of SocialCode, as shown in Figure 13. Table 6 lists normalized ratios for average file downloading delays of SocialCode, SC without DP and FRAME + CodeOn with respect to that of RCM + CodeOn under three numbers of nodes. It shows that SocialCode achieves the lowest normalized ratio for the file downloading delays among all of them for three different numbers of nodes.

Table 6. Normalized ratios for average file downloading delays of SocialCode, SC without DP and FRAME + CodeOn (FRAME) over that of RCM + CodeOn (RCM) under three numbers of nodes.

Normalized Ratio of File Downloading Delay	The Number of Nodes		
	3468	867	216
SocialCode	0.51	0.43	0.54
SC without DP	0.62	0.56	0.64
FRAME + CodeOn (FRAME)	0.80	0.80	0.81
RCM + CodeOn (RCM)	1.00	1.00	1.00

Figure 14a–c shows average credits of three types of nodes, that is, the destination, intermediate and overhearing ones, respectively, versus three numbers of nodes for SocialCode, SC without DP,

RCM + CodeOn and FRAME + CodeOn. As mentioned above, the destination node with RCM + CodeOn exhausts its credits faster than that with FRAME + CodeOn. Oppositely, when the nearby destination or overhearing node receives the innovative content piece broadcast by some destination node, SocialCode and SC without DP will stimulate this destination node by raising its average credits by α , which is paid by the receiving node.

As mentioned above, when the intermediate node with RCM + CodeOn and FRAME + CodeOn receives a content piece from the destination node, this destination one will pay one unit of its credit to increase that of the receiving intermediate node. With the same reasons mentioned in Section 4.2, SocialCode and SC without DP achieve more steady values on average credits of the intermediate node than RCM + CodeOn and FRAME + CodeOn do under three different numbers of nodes, as shown in Figure 14. Further, because SocialCode assigns dynamic forwarding priorities and backoff delays of all nearby intermediate nodes with Equations (4) and (5) respectively to schedule their sequences on re-broadcasting content pieces, the intermediate node with SocialCode has higher probability to re-broadcast an innovative content piece to its successive destination node than that using SC without DP does. Hence, the average credit of the intermediate node with SocialCode decreases slower than that using SC without DP. As shown in (a–c) of Figure 14, SC without DP gives higher average credits to the destination node but lower ones to the intermediate node than SocialCode does, no matter which number of nodes is used in the simulation. Further, the larger the total number of nodes is, the less stable and lower average credits the intermediate node has but the higher ones the destination node does. As mentioned above, these results also exhibit that SocialCode provides more even forwarding opportunities among intermediate nodes than SC without DP does.

Finally, average credits of the overhearing node using RCM + CodeOn and FRAME + CodeOn remain unchanged but those of the overhearing node using SocialCode and SC without DP reduce to zero soon. Consequently, SocialCode and SC without DP penalize the overhearing node by preventing it from receiving any successive piece or broadcasting its own ones because of its selfish operation during the CCD procedure.

6. Conclusions

In the CCD procedure of SocialCode, the vehicular node is proposed to interchange its content bitmap with nearby nodes such that it is able to calculate the correct number of various SLNC-coded content pieces, the backoff time and the number of re-encoded pieces. Then according to selfishness and interests of vehicular nodes on each content piece, SocialCode classifies them into four types. It further proposes associated SLNC-based re-broadcast operations and credit-based incentive stimulations for them to achieve efficient cooperative content distribution. Finally, using the urban map of Taipei, Taiwan, we perform NS-2 simulations, with respect to different file sizes and total numbers of vehicular nodes, for SocialCode, SC without DP, RCM + CodeOn and FRAME + CodeOn. The simulation results present that SocialCode achieves the best performance on average decoding percentage, downloading delay and credit among all four schemes for stimulating incentives of vehicular nodes to join the SLNC-based CCD process. As the best knowledge we have, SocialCode is one of the first few approaches to integrate the credit-based incentive protocol into the SLNC-based CCD process. In the future, we will first modify SocialCode to support more dynamic and practical VANET scenarios, for example, the CCD process initiated by any vehicular node, not only by the RSU in this paper. Second, we will consider security and privacy issues of SocialCode to avoid attacks like exchanging cheating messages between vehicular nodes.

Author Contributions: I.-C.C. and J.L. conceived and designed the experiments; J.L. performed the experiments; I.-C.C. and C.-E.Y. analyzed the data; I.-C.C. and C.-E.Y. wrote the paper.

Funding: This work was funded by the National Science Council (NCS), Taiwan, under Grant Number NCS102-2221-E-018-017-.

Conflicts of Interest: The authors declare no conflict of interest. The founding sponsors had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript and in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

AoI	Area of interest
CCD	Cooperative content distribution
CCH	Control channel
CCS	Credit clearance service
CR	Data communication range
DP	Dynamic priority
DTN	Disruption tolerant network
ER	Energy detection range
FRAME	Fair reimbursement and motivating sweepstake scheme
MANET	Mobile ad-hoc network
MDR	Multiplicative decreasing reward
NC	Network coding
OBU	On board units
RCM	Receipt counting method
RIS	Reciprocal incentive scheme
RSU	Road side unit
SC	SocialCode
SLNC	Symbol-level network coding
SUMO	Simulation of urban mobility
TCCS	Trusted credit clearance service
TTL	Time-to-live
V2I	Vehicle-to-infrastructure
V2V	Vehicle-to-vehicle
VANET	Vehicular ad hoc network

References

- Jiang, D.; Taliwal, V.; Meier, A.; Holfelder, W.; Herrtwich, R. Design of 5.9 Ghz DSRC-based vehicular safety communication. *IEEE Wirel. Commun.* **2006**, *13*, 36–43. [[CrossRef](#)]
- DSRC: The Future of Safer Driving. Available online: http://www.its.dot.gov/factsheets/dsrc_factsheet.htm (accessed on 15 September 2018).
- Jiang, D.; Delgrossi, L. IEEE 802.11p: Towards an international standard for wireless access in vehicular environments. In Proceedings of the IEEE Vehicular Technology Conference, Singapore, 11–14 May 2008.
- IEEE trial-use standard for wireless access in vehicular environments (WAVE)—Multi-channel operation. *IEEE Std 1609.4-2006* **2006**, c1–74. [[CrossRef](#)]
- Mitropoulos, G.K.; Karanasiou, I.S.; Hinsberger, A.; Aguado-Agelet, F.; Wieker, H.; Hilt, H.-J.; Mammari, S.; Noecker, G. Wireless local danger warning: Cooperative foresighted driving using intervehicle communication. *IEEE Trans. Intell. Transp. Syst.* **2010**, *11*, 539–553. [[CrossRef](#)]
- Huang, W.; Wang, L. ECDS: Efficient collaborative downloading scheme for popular content distribution in urban vehicular networks. *Comput. Netw.* **2016**, *101*, 90–103. [[CrossRef](#)]
- Gong, H.; Yu, L. Content downloading with the assistance of roadside cars for vehicular ad hoc networks. *Mob. Inf. Syst.* **2017**, *2017*, 4863167. [[CrossRef](#)]
- Das, S.; Nandan, A.; Pau, G.; Sanadidi, M.Y.; Gerla, M. SPAWN: A swarming protocol for vehicular ad hoc networks. In Proceedings of the 1st ACM VANET, Philadelphia, PA, USA, 1 October 2004.
- Nandan, A.; Das, S.; Pau, G.; Gerla, M.; Sanadidi, M.Y. Co-operative downloading in vehicular ad-hoc wireless networks. In Proceedings of the WONS '05, St. Moritz, Switzerland, 19–21 January 2005; pp. 32–41.

10. Nandan, A.; Tewari, S.; Das, S.; Gerla, M.; Kleinrock, L. AdTorrent: Delivering location cognizant advertisements to car networks. In Proceedings of the WONS '06, Les Ménuires, France, 18–20 January 2016; pp. 203–212.
11. Li, S.-Y.R.; Yeung, R.W.; Cai, N. Linear network coding. *IEEE Trans. Inf. Theory* **2003**, *49*, 371–381. [[CrossRef](#)]
12. Gkantsidis, C.; Rodriguez, P. Network coding for large scale content distribution. In Proceedings of the IEEE INFOCOM, Miami, FL, USA, 13–17 March 2005; pp. 2235–2245.
13. Ahmed, S.; Kanhere, S.S. Vanetcode: Network coding to enhance cooperative downloading in vehicular ad-hoc networks. In Proceedings of the IWCMC '06, Vancouver, BC, Canada, 3–6 July 2006; pp. 527–532.
14. Lee, U.; Park, J.-S.; Yeh, J.; Pau, G.; Gerla, M. Codetorrent: Content distribution using network coding in VANET. In Proceedings of the MobiShare '06, Los Angeles, CA, USA, 25–25 July 2006; pp. 1–5.
15. Lee, S.H.; Lee, U.; Lee, K.W.; Gerla, M. Content distribution in VANETs using network coding: The effect of disk I/O and processing O/H. In Proceedings of the IEEE SECON '08, San Francisco, CA, USA, 16–20 June 2008; pp. 117–125.
16. Park, J.S.; Lee, U.; Oh, S.Y.; Gerla, M.; Lun, D.S. Emergency related video streaming in VANET using network coding. In Proceedings of the ACM VANET'06, Los Angeles, CA, USA, 29 September 2006; pp. 102–103.
17. Katti, S.; Katabi, D.; Balakrishnan, H.; Medard, M. Symbol-level network coding for wireless mesh networks. *ACM SIGCOMM Comput. Commun. Rev.* **2008**, *38*, 401–412. [[CrossRef](#)]
18. Li, M.; Yang, Z.; Lou, W. CodeOn: Cooperative popular content distribution for vehicular networks using symbol level network coding. *IEEE J. Sel. Areas Commun.* **2011**, *29*, 223–235. [[CrossRef](#)]
19. Yang, Z.; Li, M.; Lou, W. CodePlay: Live multimedia streaming in VANETs using symbol-level network coding. *IEEE Trans. Wirel. Commun.* **2012**, *11*, 3006–3013. [[CrossRef](#)]
20. Yu, T.-X.; Yi, C.-W.; Tsao, S.-L. Rank-based network coding for content distribution in vehicular networks. *IEEE Wirel. Commun. Lett.* **2012**, *1*, 368–371. [[CrossRef](#)]
21. Zhu, W.; Li, D.J.; Saad, W. Multiple vehicles collaborative data download protocol via network coding. *IEEE Trans. Veh. Technol.* **2015**, *64*, 1607–1619. [[CrossRef](#)]
22. Toh, C.K.; Kim, D.; Oh, S.; Yoo, H. The controversy of selfish nodes in ad hoc networks. In Proceedings of the 12th International Conference on Advanced Communication Technology (ICACT '10), Phoenix Park, Korea, 7–10 February 2010; Volume 2, pp. 1087–1092.
23. Lu, R.; Lin, X.; Zhu, H.; Zhang, C.; Ho, P.H.; Shen, X. A novel fair incentive protocol for mobile ad hoc networks. In Proceedings of the IEEE WCNC, Las Vegas, NV, USA, 31 March–3 April 2008; pp. 3237–3242.
24. Buttyan, L.; Dora, L.; Felegyhazi, M.; Vajda, I. Barter-based cooperation in delay-tolerant personal wireless networks. In Proceedings of the IEEE International Symposium on World of Wireless, Mobile and Multimedia Networks, Espoo, Finland, 18–21 June 2007; pp. 1–6.
25. Lee, S.; Pan, G.; Park, J.; Gerla, M.; Lu, S. Secure incentives for commercial ad dissemination in vehicular networks. In Proceedings of the ACM MobiHoc, Montreal, QC, Canada, 9–14 September 2007.
26. Li, F.; Wu, J. FRAME: An innovative incentive scheme in vehicular networks. In Proceedings of the IEEE ICC, Dresden, Germany, 14–18 June 2009; pp. 1–6.
27. Chen, B.; Chan, M.C. MobiCent: A credit-based incentive system for disruption tolerant network. In Proceedings of the IEEE INFOCOM, San Diego, CA, USA, 14–19 March 2010; pp. 1–9.
28. Zhao, G.; Chen, M.; Wei, X. RIS: A reciprocal incentive scheme in selfish opportunistic networks. *Wirel. Pers. Commun.* **2013**, *70*, 1711–1734. [[CrossRef](#)]
29. Liu, H.; Lee, P.C.; Lui, J.C.S. On the credit evolution of credit-based incentive protocols in wireless mesh networks. *Comput. Netw.* **2013**, *57*, 3327–3343. [[CrossRef](#)]
30. Seregina, T.; Brun, O.; El-Azouzi, R.; Prabhu, B.J. On the design of a reward-based incentive mechanism for delay tolerant networks. *IEEE Trans. Mob. Comput.* **2017**, *16*, 453–465. [[CrossRef](#)]
31. Jethawa, H.; Madria, S. Reputation and credit based incentive mechanism for data-centric message delivery in DTNs. In Proceedings of the 19th IEEE International Conference on Mobile Data Management (MDM), Aalborg, Denmark, 25–28 June 2018; pp. 207–216.
32. Li, J.; Wang, X.; Yu, R. Reputation-based incentives for data dissemination in mobile participatory sensing networks. *Int. J. Distrib. Sens. Netw.* **2015**, *11*, 172130. [[CrossRef](#)]
33. Katmada, A.; Satsiou, A.; Kompatsiaris, I. A reputation-based incentive mechanism for a crowdsourcing platform for financial awareness. In Proceedings of the International Workshop on the Internet for Financial Collective Awareness and Intelligence (IFIN 2016), Florence, Italy, 12 September 2016; pp. 57–80.

34. Pal, S.; Saha, B.K.; Misra, S. Game theoretic analysis of cooperative message forwarding in opportunistic mobile networks. *IEEE Trans. Cybern.* **2017**, *47*, 4463–4474. [[CrossRef](#)] [[PubMed](#)]
35. Zhan, Y.; Xia, Y.; Zhang, J.; Wang, Y. Incentive mechanism design in mobile opportunistic data collection with time sensitivity. *IEEE Int. Things J.* **2018**, *5*, 246–256. [[CrossRef](#)]
36. Chang, I.-C.; Lo, J. A credit-based incentive protocol for stimulating network-coded cooperative content distribution in VANET. In Proceedings of the Eighth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS-2014), Birmingham, UK, 2–4 July 2014; pp. 452–457.
37. Simulation of Urban Mobility (SUMO). Available online: <http://sumo.sourceforge.net> (accessed on 15 September 2018).
38. NS-2. Available online: <http://www.isi.edu/nsnam/ns> (accessed on 15 September 2018).
39. Panahi, F.H.; Ohtsuki, T. Analytical modeling of cognitive heterogeneous cellular networks over Nakagami-m fading. *EURASIP J. Wirel. Commun. Netw.* **2015**, *2015*, 61. [[CrossRef](#)]



© 2018 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/>).