applied sciences

MDPI

Article

A Reliable Transport Scheme for Human Opportunistic Networks

Meihua Liu [1], Mao Tian [1,*], Xiaoli Chen [1] and Jianbin Wu [2]

[1] Electronic Information School, Wuhan University, Luoyu Road, Wuchang District, Wuhan 430072, China; gaoshz@whu.edu.cn (M.L.); cxl@whu.edu.cn (X.C.)

[2] School of Physical Science and Technology, Huazhong Normal University, Luoyu Road, Hongshan District, Wuhan 430079, China; wujianbin@phy.ccnu.edu.cn

* Correspondence: mtn@whu.edu.cn; Tel.: +027-687-78-019

Received: 8 September 2020; Accepted: 21 September 2020; Published: 23 September 2020

Abstract: The human opportunistic networks (ONs) formed by hand-held smart devices can facilitate peer-to-peer communication when humans are on the move, despite contemporaneous end-to-end paths rarely existing. In some scenarios, where network resources, especially power, are scarce and the traffic is large, the data delivery is prone to poor user experience and unbounded delay, although the best effort mechanism "store-carry-forward" is used. To cope with that, most transport/routing schemes obtain an acceptable latency at the cost of energy resources. In real-life human ONs, however, excessive energy consumption will trigger passive participation of the relays in message forwarding, so as to save their limited energy resource. Thus, the reliability of these schemes may get worse in real-life human ONs. In this paper, a reliable transport scheme is developed by making an optimal trade-off between the file round-trip delay and the energy consumption of relays. We make use of acknowledgements and coding at the source to enable successful file delivery. When setting up the network model, the cache management rule referred as "full-duplex" strategy is formulated, and then a mathematical model is established to analyze the proposed scheme. This model describes the evolution of packet dissemination and allows both the mean file round-trip delay and the energy consumption up to the reception of the last acknowledgement by the source to be expressed. Subsequently, a new function based on these two metrics is proposed to reflect the number of files that can be delivered under time and energy constraints. Through optimization procedure, the configurations that can maximize the function are obtained; thus, the optimization of these two metrics is achieved. Numerous simulations and comparisons are conducted and the results verify the accuracy of the analytical model. Comparison results show that with limited energy and passive relays, the proposed transport scheme can significantly reduce the energy consumption of file delivery, which obviously alleviates the selfish behavior of nodes. Therefore, the reliability and stability of the communication service in human ONs are enhanced.

Keywords: human opportunistic networks; reliable transport; analytical models; energy consumption

1. Introduction

Recently, the rapid proliferation of intelligent wireless devices (e.g., mobile sensors, smartphones) provides opportunities for peer-to-peer communication. In practical applications, a contemporaneous end-to-end path does not always exist due to limited wireless radio range, noise, or lack of stable electrical power supply. Under such circumstances, communication protocols of mobile ad hoc networks (MANETs) are not very efficient [1]. As an important evolution of MANETs, opportunistic networks (ONs) aim at making communication between the source and destination in an intermittently connected environment, thus becoming an attractive research area [2,3]. In ONs, a source relies on

Appl. Sci. **2020**, *10*, 6658; doi:10.3390/app10196658

the mobility of other nodes which act as relays, and takes advantage of the opportunity of contacts which occur when the mobile relays come into the radio range of each other. Data are delivered to the destination in a "store-carry-forward" manner, where a message could be sent over an existing link, getting buffered at the current node until the next link in the path comes up, until it arrives at its destination [4]. Although ONs are not competitors for the fastest or most reliable network, they have some advantages. First, they are second to none in terms of delay and interruption tolerance. In addition, they operate independently of infrastructures, can extend the cellular networks' coverage, and offload part of the wireless traffic at a low cost. These features make them very useful in many applications such as the internet of things, field military networks, interplanetary networks, and underwater sensor networks.

Opportunistic networks can be formed by intelligent wireless devices carried by human beings, such as mobile phones and laptops, to enable individuals to exchange information with each other, using this peer-to-peer opportunistic communication [5]. This kind of ON has been applied in several scenarios, for example, temporary community [6], the post-disaster system [7], vehicle ad hoc networks [8], etc. In a temporary community, people with the same purpose or interest likely encounter each other frequently; they will gather in conferences, squares, office buildings, or campuses, to construct an ON. The Wi-Fi and Bluetooth interfaces of their automobiles or iPads are utilized to share pictures, audio files, video clips, or advertisements among interested mobile users. In a post-disaster system where cellular and other traditional communication infrastructures are overused or considerably destroyed by natural disasters, rescue teams and survivors can form an ON for key information exchange through their active physical movement. In these human ONs, most of the devices are hand-held and run primarily on alternative power sources (battery) [9], so their network resources such as cache and energy are limited. In addition, the traffic is generally large: People in a temporary community tend to share files frequently with others [10]; rescue workers in post-disaster scenario would try to send reports about the affected area such as temperatures, pollution levels, and triage information of the victims to the accessible control center or the internet, and the survivors tend to connect to their relatives and friends to enquire their safety. It is of great importance to realize reliable data transport, especially in the post-disaster system, ensuring that critical information about the victims and rescue do reach the destination, which can significantly improve the success rate of rescue.

However, the exchange of messages is done based on the transient and opportunistic contacts between moving users, such that best effort networks are prone to unbounded delivery delays and poor user experience [11]. A number of reliable transport schemes for ONs have been designed for deep space communication [12–14]. These schemes make use of retransmission to cope with the unique characteristics of deep-space links, where the traffic is generally light, high data delivery rates are emphasized, and long delays are acceptable. But for the scenarios we consider, node densities are generally higher than those in deep space ONs, with higher traffic and much lower tolerance for delay. These transport protocols will incur network congestion and unacceptable latency when a huge traffic burst emerges in human ONs.

For terrestrial ONs, protocols based on rateless coding (network coding or erasure coding such as Luby transform (LT) codes and Reed-Solomon (RS) codes) are widely considered as a promising solution to enhance the reliability of ONs [15–17], thus getting more and more attention. With coding, the successful reception of a file no longer depends on receiving specific packets, but on receiving a sufficient number of individual packets, whereas without coding, the loss of a packet has to be compensated by retransmitting the same packet. Thus, the coding-based protocols can offer resilience to link failures, increase throughput, and reduce the delivery delay [18,19]. Several coding-based schemes have shown that deterministic forms of reliable transport in ON can be allowed by integrating receivers' acknowledgements (ACKs) [20,21]. Nevertheless, these solutions rely on considerable redundancy/forwarding to cater the packet losses in the network and increase the probability of the message arriving at its destination, thus obtaining acceptable latency at the cost of energy resources [22].

Therefore, they are well suited for some scenarios where network resources such as cache and energy are rich to use. However, they make the assumption that all relays are willing to participate in the routing process, which is not necessarily the case in real-life human ONs. To be specific, since network resources in realistic human society are limited, relays tend to be selfish when they are low on these resources, which we refer to as selfish nodes. A selfish node may choose not to receive packets from other nodes in order to save its resources for future use (such as energy-conserving sleep schedules). Consuming a lot of energy/cache to get a satisfactory delay will trigger the selfish behavior of considerable relays, which can lead to the noncooperation of relays. As a result, the performance of these schemes will be degraded. Therefore, to enhance the reliability of real-life human ONs, the transport scheme should take file delivery delay as well as the resource consumption into account.

There are a few transport schemes for ONs to date to consider the resource consumption in the minimization of delay. Ali et al. [23] proposed a reliable transport protocol based on coding and ACKs aiming at minimizing the round-trip file delay by controlling the allocation of the limited cache for source packets and ACKs dynamically. When constructing the network model, they assumed that only one packet can be stored per node, and adopted the cache management rule in which the source packets and acknowledgements competed for the rare cache, which was defined as the "half-duplex" feature. However, they neglected the influence of continuous energy consumption on the network performance. Although the authors proposed an improved version which took the energy consumption of the network into account, they still focused on optimizing the delivery delay by allocating the limited cache space, and simply setting an upper limit on energy consumption [24]. Therefore, an optimal trade-off between energy consumption of relays and the delivery delay remains to be further studied. In this paper, our goal is to develop a more reliable transport scheme by achieving the optimal trade-off between the two metrics: We make use of acknowledgements and coding at the source. For the sake of concentrating on the influence of energy consumption on the performance of the scheme, we adopt a loosen cache management rule that there is no contention between the source packets and acknowledgements for storage, which is defined as the "full-duplex" strategy. Then an analytical model is set up to describe the evolution of packets dissemination. Based on the model, a new function involving the round-trip file delay and the energy consumption is proposed to reflect the number of files that can be delivered under time and energy constraints, in order to comprehensively evaluate the impact of these two metrics on the transmission efficiency. By maximizing the function, the optimal network configuration parameters can be obtained, achieving the best compromise between delay and energy consumption, thus improving the transmission efficiency and reliability. The main contributions of this paper are threefold:

1. We propose a reliable transport scheme for real-life human ONs, which optimizes the delay of the source file and the energy consumption of relays. The scheme makes use of acknowledgements and coding at the source to enable successful file delivery. A cache management rule referred as the "full-duplex" strategy is formulated and a mathematical model is set up to analyze the proposed scheme;
2. A new function based on round-trip file delay and the corresponding energy consumption is proposed to reflect the number of files that can be delivered under time and energy constraints. Through optimization procedure, the configurations that can optimize these two metrics are obtained;
3. The accuracy of the analytical model is verified by numerical simulations, and the performance of the proposed reliable transmission mechanism is evaluated. Specifically, the performance metrics we consider are the energy consumption and the mean round-trip delay of the file (until the final acknowledgement reaches the source). The remainder of this paper is organized as follows: We discuss related work in Section 2. In Section 3, the network setting is described. In Section 4, a detailed account of the proposed transport scheme is presented. In Section 5, an analytical model is developed to analyze the proposed scheme, and a new function based on round-trip file delay and the corresponding energy consumption is proposed and optimized to make a good

trade-off between the two metrics. In Section 6, we demonstrate the accuracy of the analytical model, optimization procedure, and the performance of the proposed scheme. Finally, Section 7 concludes this work and provides future directions of research.

2. Related Work

As is known to all, TCP is the most widely used reliable transport protocol. But it is proved to be inefficient in ONs, since it misinterprets losses due to route failures and losses due to network congestion [25,26]. This is even worse in the case of intermittently connected ONs. Transport protocols for MANETs are mainly based on cross-layer signaling between routing and transport so as to notify the transport layer about route failures [27,28]. They are not suitable for ONs because only opportunistic routing can be performed.

A number of reliable transport schemes in ONs are designed for deep space communication, where the traffic is generally light, high data delivery rates are emphasized, and long delays are tolerant. These schemes aim at extending the TCP mode of operation to cope with the unique characteristics of deep-space links. For example, literatures' [12–14] protocols use retransmission mechanism to achieve transmission reliability, which means that the recovery delay of a lost packet needs to wait at least one supplementary round-trip time (RTT), and the end-to-end delay will be large. This is not problematic for deep space opportunistic networks. However, human ONs have a relatively larger number and higher density of nodes compared to deep-space ON, with more urgent demand for data exchange and lower tolerance for transmission delay; these transport protocols will incur network congestion and unacceptable latency when a huge traffic burst emerges in human ONs with resource constraints. Other solutions such as Licklider transmission protocol (LTP) [29] and those offered by Saratoga [30] make improvements on the basis of the Buddle Protocol (the IETF RFC 5050) [31] proposed by the DTN research group. Specifically, using automatic repeat reQuest (ARQ) on a hop-by-hop basis, they are able to deliver only a per-hop reliability. In addition, the end-to-end delay still depends on the RTT and may exceed the range in human ONs. For terrestrial ONs, to reduce delay and improve the storage efficiency, rateless coding has been increasingly applied in transport schemes for ONs [16,17,32]. With coding, the successful reception of a file no longer depends on receiving specific packets, but on receiving a sufficient number of individual packets. Thus, the coding-based protocols can offer resilience to link failures, as well as increase throughput and reduce the delivery delay.

Several coding-based schemes have shown that reliable transmission in ON can be allowed by integrating receivers' acknowledgements (ACKs). Zeng et al. proposed a reliable transmission mechanism for streaming data by using acknowledgements and network coding [33]. Their pipeline-based stop-and-wait transmission mechanism makes full use of the waiting time for ACK, significantly reduces the data transmission delay, and improves the decoding efficiency. Thai et al. analyze re-encoding strategies at intermediate nodes using an on-the-fly coding paradigm, and propose an adaptive mechanism based on acknowledgements and coding to enable end-to-end reliable communication [34]. Simulation results show that their adaptive mechanism can ensure end-to-end reliability and allow a significant reduction in terms of network overhead. However, they inject considerable redundancy into the network to obtain acceptable performance at the cost of energy consumption at the relays, and they ignore the fact that excessive energy consumption of relays will influence the relays' behavior. In real-life human ONs, consuming a lot of energy will lead to the selfish behavior of relays, and then the reliability of file transfer will be affected. Therefore, to encourage more relays to participate in the transfer process, a reliable transport scheme in real-life human ONs has to take the energy consumption of relay nodes into consideration.

Although there are several energy-efficient communication protocols for ONs [35], a few transport protocols considering energy consumption have been proposed in recent years. Sassatelli et al. propose a new reliable transport scheme based on coding and ACK, aiming at optimizing round-trip delay of file under energy constraints [23,24]. The transmission is organized within cycles. During each cycle, the source transmits a number of redundant packets based on the last ACK received from the destination, and then stops transmitting packets to wait for the ACK. They assume that each node can only store one packet, and consider the buffer management strategy that source packets and acknowledgements have to compete for storage. A "half-duplex feature" is used to describe this contention. What they focus on is how to allocate the limited storage space of redundant packets and ACKs to minimize the delay of the file. Although the limited energy is taken into consideration, their optimization objectives are still the delivery delay of the file, and to simply give an upper bound to energy consumption. Hence, an optimal trade-off between energy consumption of relays and the delivery delay remains to be further studied. In this paper, what we focus on is how to improve the reliability of the network by reducing the network energy consumption. We assume that there is no competition between redundant packets and ACKs at the node cache, and a reliability transport scheme is proposed though the optimization of both the energy consumption and the file round-trip delay.

3. Network Model

Only unicast from source node S to destination node D was considered in this paper. The network consisted of $N + 2$ wireless mobile nodes (S, D, and N relay nodes) connected in multi-lines, and the links were time-varying. The network was assumed to be sparse: Contemporaneous end-to-end paths were not guaranteed, and the ratio between the coverage area of all nodes and the total area was low so that we neglected interference. We also assumed that two mobile nodes exchanged data via short-range interfaces (Bluetooth, WIFI, etc.), and that communications were bidirectional. Messages were buffered at each intermediate node and waited until the next hop was available, which could take a long period of time. Each node had the same and limited bandwidth and cache size. In addition, studies have shown that many popular movement models, such as random waypoints, random walks, and community-based models, have the property that the inter-contact time of nodes is exponentially distributed or has an exponential tail [36,37]. In order to guarantee the universality of our method, this paper does not assume a specific movement model, but adopts the assumption that the inter-meeting interval between node pairs are independent random variables that follow the same exponential distribution (rate parameter is λ). The proposed analytical framework is also effective for other realistic movement models.

This paper only considers coding at the source node: The file was cut into K pieces by the source node and transmitted as coded packets in cycles, where the RaptorQ was employed. The destination node could decode the source file with high probability after collecting several arbitrary coded packets more than K. To ensure that the source node could get the number of coded packets delivered in time, the destination node sent multiple ACKs in each cycle: When destination node D received a new coded packet, an ACK q was sent back to the source. The index value q reflects the number of coded packets needed by the destination currently to recover the source file.

When routing coded packets, we adopted time-limited epidemic routing, which utilized multiple possible data propagation paths from the source to the destination to reduce the latency [38]. On the feedback path of ACK, unlimited epidemic routing was used to send it to the source node as soon as possible.

3.1. Cache Management Rule

In real-life human ONs, most wireless devices have limited cache capacity. For the sake of concentrating on the investigation of the influence of energy consumption on file delay, we adopted a loose cache management rule that there was no contention between the source packets and acknowledgements for storage, which was defined as the "full-duplex" feature.

We assumed that the source and destination could accommodate all K coded packets, while the cache of each intermediate relay node could only accommodate H packets, where $1 \leq H \leq K$. Since the ACK was very small, we assumed that there was no contention between the ACK and coded package for storage, and that the ACKs would not be dropped until a new ACK overrode them. In addition, we also assumed that only one packet could be transmitted in every contact. It was easy to extend this assumption to general cases that nodes could forward any number of packets between them when encounter opportunities occurred.

3.2. RaptorQ Coding

Fountain code [39] is a kind of forward erasure code, whose main idea is to encode the data to be transmitted to form "water droplets", while destination can recover the original file with high probability by receiving enough water droplets, regardless of the specific "water droplets". Relative algorithms include LT [40], Raptor [38], and RaptorQ coding [41]. Raptor evolved from the LT code, and RaptorQ is an improved version of Raptor. The Raptor encoder generates n_1-encoding packets from k original packets ($k < n_1$), and then sends them to the destination.

Decoding failure probability [38] with n_2 received encoding packets of the RaptorQ is:

$$P'_f = \begin{cases} 1, & n_2 < k \\ 0.01 \times 0.01^{n_2-k}, & n_2 \geq k \end{cases} \tag{1}$$

Thus, the decoder can recover all the source files with a very high probability by collecting an arbitrary set of encoding packets that are only slightly more than the original packets number k. It should be noted that other coding techniques such as Reed Solomon codes, network coding, etc., can also be used in the proposed transport mechanism.

4. The Proposed Reliable Transport Scheme

4.1. An Overview of the Reliable Transport Process

File transmission was carried out in cycles; the transmission process of each cycle is shown in Figure 1. At the beginning of cycle r, the source node determined the number of packets to be injected into the network M_r; the diffusion timer $T_{s,r}$, and the waiting timer $T_{w,r}$ were based on the index value of the latest feedback ACK_{r-1} received in the previous cycle (r−1). After these packets were sent out, their relay nodes were allowed to replicate and forward the packets over the network for the duration of timer $T_{s,r}$. Then the source node waited for $T_{w,r}$ until it received an ACK from the destination. Thus, the duration of a cycle was $\frac{M_r}{\lambda} + T_{s,r} + T_{w,r}$. At the end of the cycle, the entire network was cleaned up and all packets in the relays were cleared (except ACKs) to avoid network congestion and the selfish behavior of relays caused by overuse of cache. The next cycle would continue unless the ACK received by the source node indicated that the destination node had collected K-independent coded packets and reconstructed the source file.

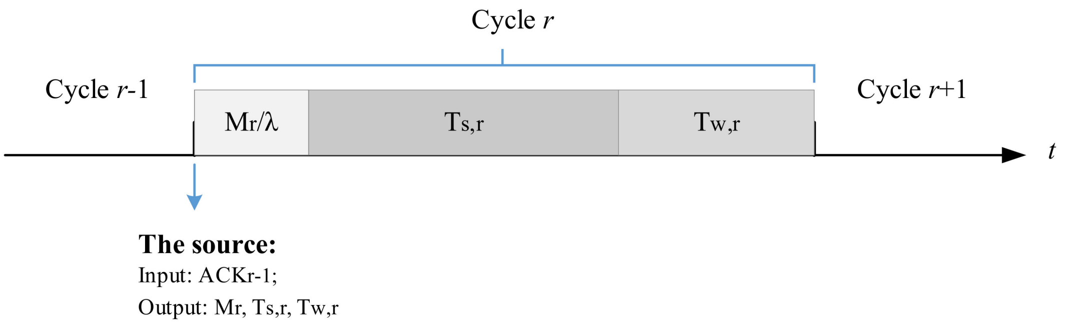


Figure 1. File transmission cycles.

It is important to note that ACKs were not cleared by the node at the end of the cycle. During the whole period of file delivery, the destination node would feedback an ACK to the network every time it received a new independently coded packet, and route it in epidemic fashion. The ACK took the number of packets still needed currently for file decoding by the destination node as the index value

q. When a relay node carrying ACK q' encountered a new ACK q, if the index value q' was greater than q, it would override ACK q' with ACK q to ensure that the latest feedback information could be transmitted in the network. Once the source node received ACK 0, the file delivery was completed and the transfer process was terminated. Otherwise, the next cycle began.

4.2. Steps of the Scheme

The transport scheme of this paper involved the following parameters: (1) The number of coded packets injected into the network by source nodes per cycle M; (2) coded packets' diffusion time T_s; (3) time period waiting for ACK return T_w. Each parameter is a function of r, which indicates the number of packets that the destination node still needs to decode source file. r was provided by the index of ACK received by the source node at the end of the last cycle, which may have differed from the actual packet collection of the destination node. The proposed reliable transport scheme was summarized in Algorithm 1.

Algorithm 1 File transport scheme

```
1.  Initialize: $r = K$;
2.  While $r > 0$
3.  Start a new cycle $r$. The source node sends $M_r$ coded packets to the network, and each time it encounters
    a relay node, the source gives it a new packet until all $M_r$ packets are sent;
4.  Packets diffusion phase: Each packet is disseminated and routed in an epidemic fashion last for time
    $M_r/\lambda + T_{s\_r}$, where $M_r/\lambda$ is the time required for the source to send out $M_r$ packets in step 3;
5.  Waiting phase for ACKs: At the end of the diffusion phase, the source node waits for ACKs (there may be
    multiple) from the destination node, and the waiting period is: $[M_r/\lambda + T_{s\_r}, M_r/\lambda + T_{s\_r} + T_{w\_r}]$. During
    this phase, coded packets are no longer copied to relays; they are only delivered to the destination node.
    The reason for setting the waiting time is that in this intermittently connected network, it takes time for
    an ACK to propagate from the destination to the source. To sum up, the length of a cycle is:
    $M_r/\lambda + T_{s\_r} + T_{w\_r}$;
6.  At the end of cycle $r$: All relays clear the coded packets in the cache. The source node takes $q$, which is the
    minimum index value of the received ACKs, as an argument for the next cycle;
7.  Update the value of $r$: $r = q$.
8.  End While
```

5. Analytical Modeling

In this section, the analysis of the proposed reliable transport process is presented. In the first and second sub-sections below, the transmission model of coded packets during one cycle and the backpropagation model of ACKs are given, respectively. Then we can model the sequence of cycles to obtain the file round-trip delay (the average delay for the source to receive ACK 0) and energy consumption defined as the total number of transmissions in the third sub-section. Based on the model, the maximization of the network performance which takes the energy consumption and round-trip delay into account can be achieved by optimizing the system parameters $\{M_r, T_{s\_r}, T_{w\_r}\}_{r=1}^{K}$, which will be presented in next section.

5.1. Dynamic Transmission Model of Coded Packets during One Cycle

We developed an analytical model to describe the dissemination process of coded packets during one cycle. Based on this model, the cumulative distribution function (CDF) of random variable D_i, the time that the destination obtains i packets, for $i = 1, \ldots, H$, was calculated.

For a cycle r, the source node injected M_r-independent-coded packets into the network. Considering that the cache of an intermediate node could store H-encoding packets at most, we divided the intermediate nodes in the network into $H + 1$ types $(0, 1, 2, \ldots, H)$ according to the actual number of coded packets stored in the cache. Nodes storing i coded packets were denoted as type v_i, where

$0 \leq i \leq H$. The receiving rate of nodes of type v_i was represented by $Re_i(t)$ ($0 \leq i \leq$ H). That is, the expected number of new independent packets received by intermediate nodes of type v_i in a unit interval and ${Re_i}^d(t)$ represented the receiving rate of the destination node.

We take $F_i(t)$ (P($D_i < t$)) as the CDF of D_i, then the derivative of $F_i(t)$ can be approximated as:

$$\frac{dF_i(t)}{dt} = \lim_{\Delta t \to 0} \frac{F_i(t+\Delta t) - F_i(t)}{\Delta t} = \lim_{\Delta t \to 0} \frac{P(D_i > t) - P(D_i > t+\Delta t)}{\Delta t} \tag{2}$$

The event $\{D_i > t + \cdot t\}$ is equivalent to $\{D_i > t\}$, then $\{D_i > t + \cdot t\}$=$\{D_i > t + \cdot t,\ \ D_i > t\}$. Therefore, we can get Equation (3):

$$P(D_i > t + \Delta t) = P(D_i > t + \Delta t, D_i > t) = P(D_i > t) \cdot [1 - P(D_i < t + \Delta t \big| D_i > t)] \tag{3}$$

For $P(D_i \langle t + \Delta t | D_i > t)$, the probability for the destination node receiving multiple coded packets in a very short interval $\cdot t$ is so low that this situation can be ignored. Then the event $\{D_i < t + \cdot t\}$ that the destination received i packets before time $t + \cdot t$ only occurs if the event $\{D_{i-1} < t\}$ that the destination received $i - 1$ packets before t happens. Therefore, we can get Equation (4):

$$P(D_i < t + \Delta t \big| D_i > t) \approx \Delta t \cdot \mathrm{Re}^d_{i-1}(t) \cdot P(D_i < t \big| D_i > t), \tag{4}$$

where ${Re_{i-1}}^d(t)$ represents the receiving rate when the destination node stores $i - 1$ coded packets, that is, the probability of receiving a new encoding packet within $\cdot t$.

Substituting Equations (3) and (4) into Equation (2), we can get:

$$\frac{dF_i(t)}{dt} = \lim_{\Delta t \to 0} \frac{\Delta t \cdot \mathrm{Re}^d_{i-1}(t) \cdot [P(D_{i-1} > t) - P(D_i > t)]}{\Delta t} = \mathrm{Re}^d_{i-1}(t) \cdot [F_{i-1}(t) - F_i(t)] \tag{5}$$

$$\frac{dF_1(t)}{dt} = \mathrm{Re}^d_0(t) \cdot [1 - F_1(t)] \tag{6}$$

The initial values are $F_i(0) = 0$, $(i = 0, 1,\ 2, \ldots, M_r)$. To obtain $F_i(t)$, ${Re_i}^d(t)$ has to be calculated.

- The calculation of $Re_i(t)$

The network state was described by the packet distribution in relay nodes by a H-tuple $\{X_0(t), X_1(t), \ldots, X_H(t)\}$. Where $X_i(t)$ represents the number of type v_i nodes at time t, and obviously $\sum_{i=0}^{H} X_i(t) = N$. In the packet diffusion phase, when a type v_i node m_i encounters a type v_j node m_j, the replication operation would not take place if a packet to be copied to m_i already exists in the cache of m_i. Therefore, there is a certain probability for node m_i to obtain a new independent packet at m_j. We denote it as P(i, j).

Since we assumed that only one coded packet could be forwarded at each encounter, the receiving rates of nodes (including intermediate nodes and the destination node) are shown in Formulas (7)–(10).

For intermediate nodes:

$$\mathrm{Re}_0(t) = \begin{cases} \lambda\left(\sum\limits_{j=1}^{H} X_j + 1\right), & 0 < t \leq T_{s\_r} \\ 0, & T_{s\_r} < t \leq T_{s\_r} + T_{w\_r} \end{cases} \tag{7}$$

$$\mathrm{Re}_i(t) = \begin{cases} \lambda\left(\sum\limits_{j=1}^{i} X_j \cdot P(i,j) + \sum\limits_{j=i+1}^{H} X_j + 1 - P(i,i)\right) & 0 < t \leq T_{s\_r} \\ 0 & T_{s\_r} < t \leq T_{s\_r} + T_{w\_r} \end{cases} \quad (i = 1, 2, \ldots, H-1) \tag{8}$$

$$\mathrm{Re}_H(t) = \begin{cases} \lambda \cdot \left(\sum\limits_{j=1}^{H} X_j \cdot P(H,j) - P(H,H) + 1 \right) & 0 < t \le T_{s\_r} \\ 0 & T_{s\_r} < t \le T_{s\_r} + T_{w\_r} \end{cases} \tag{9}$$

For destination node:

$$\mathrm{Re}_i^d(t) = \lambda \cdot \left(\sum\limits_{j=1}^{i} X_j \cdot P(i,j) + \sum\limits_{j=i+1}^{H} X_j + 1 \right) \quad (i = 0, 1, 2, \ldots, M_r - 1) \ , \\ \text{and } \mathrm{Re}_{M_r}^d(t) = 0 \tag{10}$$

During packet diffusion phase $T_{w,r}$, nodes of type v_0 could obtain a packet from any encountered node that stored packets. For nodes of v_j, there was a probability that they could obtain a new packet from an encountered node. During the waiting period T_{w_r}, all relay nodes would no longer receive any coded packet, so the receiving rates was 0. But note that the destination node could receive coded packets in all time periods. Therefore, the receiving rate of the destination node in both $T_{s,r}$ and T_{w_r} periods can be obtained from Formula (10). Apparently, $\mathrm{P}(i,j)$ and $X_i(t)$ had to be obtained to calculate the $Re_i(t)$:

- The calculation of $\mathbf{P}(i,j)$

In the packet diffusion phase, when a type v_i node m_i encounters a type v_j node m_j, the replication operation will not take place if a packet to be copied to m_i already exists in the cache of m_i. Therefore, there is a certain probability for node m_i to obtain a new independent packet at m_j, which is denoted as $\mathrm{P}(i,j)$. We assumed that the i packets carried by m_i were evenly distributed among the M_r code packets sent by the source. This assumption was reasonable if the global rarest policy, which could maintain close to even proportions of K packets in the network, was used. Thus, the calculation of $\mathrm{P}(i,j)$ can be conducted, as shown in Equation (11):

$$P(i,j) = \begin{cases} 1 & i < j \\ 1 - \frac{c_i^j}{c_{M_r}^j} & i \ge j \end{cases} \tag{11}$$

When $i < j$, m_i must get a new independent packet at m_j; otherwise, under the assumption of uniform distribution of coded packets, the probability that all packets in m_j have copies in m_i is $\frac{C_i^j}{C_{M_r}^j}$. A similar analysis has been applied to BitTorrent, such as P2P file sharing systems.

- The calculation of $X_i(t)$

$X_i(t)$ represents the number of nodes of type v_i. When the type of nodes changes from v_{i-1} to v_i after receiving a new coded packet, $X_i(t)$ increases; v_i becomes v_{i+1} after receiving a new coded packet, then $X_i(t)$ decreases. The change rate of $X_i(t)$ is related to the receiving rate of nodes in type v_j over a very short time interval. In this way, the following ordinary differential Equation (12) can be used to calculate $X_i(t)$:

$$\begin{aligned} \frac{dX_i(t)}{dt} &= \mathrm{Re}_{i-1}(t) \cdot X_{i-1}(t) - \mathrm{Re}_i(t) \cdot X_i(t) \quad (i = 1, 2, \ldots, H-1) \\ \frac{dX_H(t)}{dt} &= \mathrm{Re}_{H-1}(t) \cdot X_{H-1}(t) \end{aligned} \tag{12}$$

where $Re_i(t)$ is a function of $X_i(t)$. Then Equation (12) can be solved with the initial values $X_i(t) = 0$, for $i = 1, 2, \ldots, H-1$.

Thus, substitute $X_i(t)$ back into Equations (7)–(10), the receiving rate $Re_i(t)$ and $Re_i{}^d(t)$ can be derived. Substitute $Re_i(t)$ and $Re_i{}^d(t)$ back into Equations (5) and (6), then the $F_i(t)$, the CDF of the time that the destination obtains i packets, for i = 1, . . . , H, can be finally calculated.

The above analysis was conducted under the assumption that each node could forward at most one packet every time they met a neighbor. There may be situations where bandwidth could support multiple packets transmission in practical applications. The solution was similar to the above method, which will be taken as the future research content.

5.2. Modeling the Backpropagation Dynamics of ACK

As mentioned above, when a new coded packet was received by the destination node, an ACK q was fed back to the source by epidemic routing. ACK q would overwrite the existing ACK, which had a larger index value q'. This means that only one ACK message could be stored in the cache of node. Since ACKs in the cache of nodes would not be cleared at the end of each cycle, the backpropagation state of ACKs in current cycle would be affected by the past cycles. Therefore, we investigated the backpropagation of ACKs from a global perspective. In this section, the CDF of the D_{Y_q}, which is the delivery delay of ACK q from the destination to the source, was calculated. Hence, the probability that the source node had received ACK q by time t, denoted as $P_{Y_q}(t)$, was obtained.

Let $Y_q(t)$ represent the number of relay nodes that store ACK q at time t, and $y_q(\mathrm{t})$ represent the number of ACK q that reaches the source node at time t. $\left\{y_q(t), t \geq 0\right\}$ can be viewed as a non-homogeneous Poisson process with parameter $\lambda \cdot Y_q(t)$. It is observed that $Y_q(T)$, the number of ACK q reaches the source node in the interval [0, T], conditioned on the natural filtration $\mathcal{F}_{Y_q}$, is a random variable that obeys the Poisson distribution with parameter $\lambda \cdot \int_0^T Y_q(t)dt$. Then, the probability that the delivery delay D_Y of ACK q is greater than T, conditioned on the natural filtration $\mathcal{F}_{Y_q}$, is given by:

$$P\left(D_{Y_q} > \mathrm{T}|\mathcal{F}_{Y_q}\right) = P\left(y_q(\mathrm{T}) = 0\middle|\mathcal{F}_{Y_q}\right) = e^{-\lambda\int_0^T Y_q(t)dt} \tag{13}$$

Take the expectation from both sides of Equation (13):

$$P(D_{Y_q} > T) = E\left(e^{-\lambda\int_0^T Y_q(t)dt}\right) \tag{14}$$

According to literature [18] by Altman et al., Equation (14) is bounded for the number of all relay nodes N. Using the dominated convergence theorem, we obtain:

$$P(D_{Y_q} > T) = \lim_{N\to\infty} E\left(e^{-\lambda\int_0^T Y_q(t)dt}\right) = e^{-\lambda\int_0^T Y_q(t)dt} \tag{15}$$

Therefore, for large N, the CDF of delay D_{Y_q} of ACK q ($q = 0, 1, 2, \ldots, r-1$), $P_{Y_q}(t)$, can be approximated by the solution of the following ordinary differential Equation (16):

$$\frac{dP_{Y_q}(t)}{dt} = \lambda \cdot Y_q(t) \cdot \left(1 - P_{Y_q}(t)\right) \quad (0 < t < +\infty\ ) \tag{16}$$

with the initial condition $P_{Y_q}(0) = 0$. Where $Y_q(t)$ can be obtained by solving the following ordinary differential equation:

$$\frac{dY_q(t)}{dt} = \lambda \cdot Y_q(t) \cdot \left(N - \sum_{j=1}^{r} Y_j(t)\right) + \lambda \cdot S_{q\_r}(t) \cdot \left(N - Y_q(t)\right) + \lambda \cdot Y_q(t) \cdot \sum_{m>q} Y_m(t) - \lambda \cdot Y_q(t) \cdot \sum_{m<q} Y_m(t) \tag{17}$$

with the initial condition: $Y_q(0) = 0$, $q = 0, 1, 2, \ldots, r-1$, where $S_{q\_r}(t)$ represents the probability that the destination node has collected r-q-coded packets at time t within the current cycle. It can be obtained from the $F_i(t)$ which was calculated in the Equations (5) and (6):

$$S_{q\_r}(t) = F_{r-q}(t) - F_{r-q+1}(t) \tag{18}$$

To understand Equation (18), the first item on the right of the equal sign indicates that the ACK q is copied by intermediate carriers to nodes without any ACK stored in unit time; the second item indicates the diffusion of ACK q by the destination node, and the $\lambda \cdot Y_q(t) \cdot \sum_{m>q} Y_m(t)$-$\lambda \cdot Y_q(t) \cdot \sum_{m<q} Y_m(t)$ indicates the replacement of the existing ACKs with the new ACKS.

5.3. Modeling the Sequence of Cycles

We now model the transport scheme by accounting for the sequence of cycles in order to get the expected completion time (round-trip delay) of the file and the energy consumption. Let q_n ($q_n \geq 1$) denote the number of coded packages still absent for the destination node to decode source file at the beginning of nth cycle. It is obvious that $\{q_n, q_n \geq 1\}$ can be seen as a Markov chain with state space $\{0, 1, 2, \ldots, K\}$. The chain begins with $q_1 = K$, and gets absorbed in state 0. Let $P_{rj}(t)$ represent the transfer probability of q_r from state r to state j. That is, the lowest-index ACK received by the source node at t is j, given that the source starts up with cycle r. $P_{rj}(t)$ can be calculated according to the Formula (19):

$$P_{rj}(t) = P_{Yj}(t) \cdot \prod_{v=0}^{j-1} (1 - P_{Yv}(t)), \text{ for } r \neq j, \tag{19}$$

and

$$P_{rr}(t) = 1 - \sum_{j=0}^{r-1} P_{rj}(t), \tag{20}$$

where $P_{Y_j}(t)$ is the probability that the source node has received ACK j at t.

- Expected round-trip delay of file

Let ED_r ($r = 1, 2, \ldots, K$) represent the expected time to reach the final state 0 from the beginning of cycle r. Thus, ED_K represents the expected round-trip delay for the source to receive ACK 0. ED_r can be calculated by the following formula:

$$ED_r = P_{r0}(T_r) \cdot E[T_{r \to 0}] + (1 - P_{r0}(T_r)) \cdot T_r + \sum_{j=1}^{r} P_{rj}(T_r) \cdot ED_j \tag{21}$$

where T_r is the length of cycle r, and $T_r = T_{s\_r} + T_{w\_r}$. $E[T_{r \to 0}]$ denotes the expected time to reach state 0 by the end of the current cycle r, given that the source starts up with cycle r. $E[T_{r \to 0}]$ can be calculated by the following formula:

$$E[T_{r \to 0}] = \int_0^{T_r} t \cdot f_{r \to 0}(t) dt = T_r \cdot F_{r \to 0}(T_r) - \int_0^{T_r} F_{r \to 0}(t) dt \tag{22}$$

where $f_{r \to 0}(t)$ and $F_{r \to 0}(T_r)$ are the probability distribution function (PDF) and CDF of ACK 0 delay, under the condition that this delay is lower than T_r. Thus, we have $F_{r \to 0}(t) = \frac{P_{r0}(t)}{P_{r0}(T_i)}$. Substituting $E[T_{r \to 0}]$ into Formula (21), we can get the final expression of ED_r:

$$ED_r = \frac{T_r - \int_0^{T_r} P_{r0}(t) dt + \sum_{j=1}^{r-1} P_{rj}(T_r) \cdot ED_j}{1 - P_{rr}(T_r)}, \quad (r = 1, 2, \ldots K) \tag{23}$$

- Expected energy consumption

According to literature [42], the energy consumption of a network is mainly related to the total forwarding times of messages. For the sake of simplicity, we define it as the product of the unified cost of a transmission and the total number of packet transmissions. Let EC_r ($r = 1, 2, \ldots, K$) represent the expected energy consumption to reach the final state 0 starting from the beginning of cycle r. Similar to the derivation of ED_r, we can express the EC_r:

$$EC_r = C_{trans} \cdot \frac{P_{r0}(T_r) \cdot \sum_{i=1}^{H} X_i(E[T_{r \to 0}]) + (1 - P_{r0}(T_r)) \cdot \sum_{i=1}^{H} X_i(T_r) + \sum_{j=1}^{r-1} P_{rj}(T_r) \cdot EC_j}{1 - P_{rr}(T_r)} \\ (r = 1, 2, ..., K) \tag{24}$$

where C_{trans} denotes the unified cost of a transmission, and $X_i(t)$ is the number of type v_i nodes at time t. From the above expressions, ED_K and EC_K are functions of parameters $\{M_r, T_{s\_r}, T_{w\_r}\}$ $(r \in (1, 2, \ldots, K))$. Thus, if the network knowledge λ and N, as well as parameters $\{M_r, T_{s\_r}, T_{w\_r}\}$, $(r \in (1, 2, \ldots, K))$ is obtained, we can get the expected round-trip delay ED_K and the corresponding energy consumption EC_K.

5.4. The Proposed Function and Its General Optimization Procedure

The goal of our reliable transport scheme was to optimize the round-trip delay of the file and the corresponding energy consumption. In this section, we propose a new function which combined the round-trip delay and energy consumption, and optimized it using general optimization algorithms.

Let TE denote the estimated value of total energy of the network, and DC denote the duration of the communication service. In some applications, TE and DC are priori information that can be obtained in advance. For example, in an ON formed by scholars attending a conference, the duration of the communication service is the length of the conference. We can estimate the range of total energy from the number of participants (the wireless devices are mostly mobile phone, equipped with 1500–2500 mAh lithium batteries). Similarly, under an emergency scenario, the ON is formed by the rescue team and survivors. The best rescue time can be seen as the duration of the communication service, and the range of total energy can be estimated by the permanent population and popularity of mobile phones of the affected area, as well as the energy of devices carried by rescue works. Therefore, we assume that TE and DC are known.

The proposed metric is expressed as follows:

$$NDM = \alpha \cdot \frac{TE}{EC_K} + (1 - \alpha) \cdot \frac{DC}{ED_K} \tag{25}$$

Recall that EC_K and ED_K are the mean energy consumption and round-trip delay of a file, respectively. To understand Equation (25), $\frac{TE}{EC_K}$ can be interpreted as the mean number of files that can be delivered under the current energy of the network; similarly, $\frac{DC}{ED_K}$ can be interpreted as the mean number of files that can be delivered during the period when the network provides communication services. With this interpretation, the NDM can reflect the average number of successful deliveries that the current opportunity network can support. By setting the weight parameter α ($\alpha \in [0,1)$), users can adjust the contributions of energy consumption and latency to the network performance, respectively.

To achieve an optimal trade-off between energy and delay, we maximized NDM by selecting the optimal parameters $\{M_r, T_{s\_r}, T_{w\_r}\}$, $(r \in (1, 2, \ldots, K))$. There are many general algorithms that have been developed to solve the above nonlinear optimization problems. In this paper, a simple and fast stochastic optimization method with good robustness—the differential evolution algorithm—was used [43]. The details of the algorithm are not be described here.

It is important to note that the above optimization process does not need to be performed on the source node in real time, but in an offline manner. All possible cycles r (r = 1, 2,..., K) are calculated and then stored in the memory of source node for actual use.

6. Simulation and Analysis

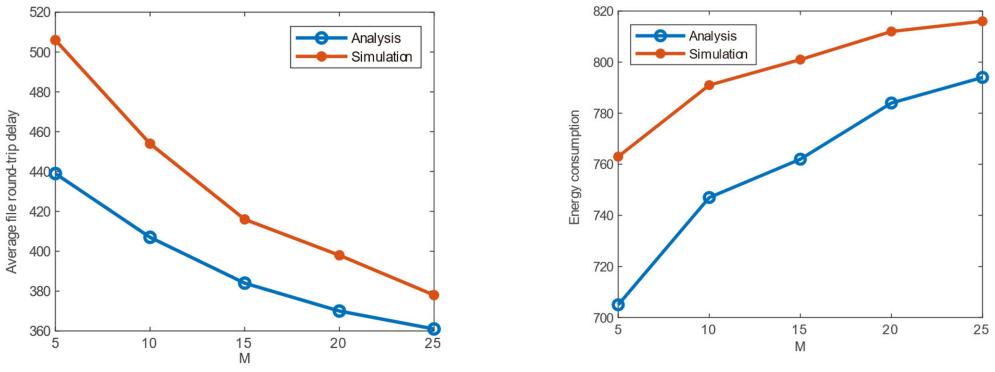
In this section, the proposed mathematical model is verified at first, then the performance of the optimal procedure is presented, and the performance of the proposed transport scheme is investigated finally. The simulations were conducted on a discrete event simulator—The ONE. As for coding and routing strategy, RaptorQ coding and time-limited epidemic routing were implemented. The inter-meeting interval between each node pair was generated using exponential distribution with rate parameter λ. Several key parameters were set as follows: N = 120, K = 5, the number of packets that a node could accommodate H was set to 2, and $\lambda = 0.01$ was obtained by a random movement model developed in MATLAB, which was then imported in The ONE. Note that the unit of λ was not specified, since λ could be 0.01 s^{-1}, 0.01 min^{-1}, or 0.01 h^{-1}. The average over 100 independent experiments was taken as the result.

6.1. Validation of the Analytical Model

In Sub-Section 5.3, we set up an analytical model to calculate the mean file round-trip delay EC_K and the corresponding energy consumption ED_K. To validate the accuracy of this model, we varied the parameters $\{M_r, T_{s\_r}, T_{w\_r}\}_{r=1}^{K}$ and assessed numerically the fit between the theoretical and simulation results. Meanwhile, we also investigated the impact of each parameter on the EC_K and ED_K. To simplify the analysis, we restricted the value of parameters M_r, T_{s_i}, T_{w_i} to be constant, that is, $M_r = M$, $T_{s\_i} = T_s$, $T_{w\_i} = T_w$ over all cycles i = 1,2,...,K. The parameters we varied were M, T_s, and T_w. When analyzing the impact of one parameter, we set the other two fixed.

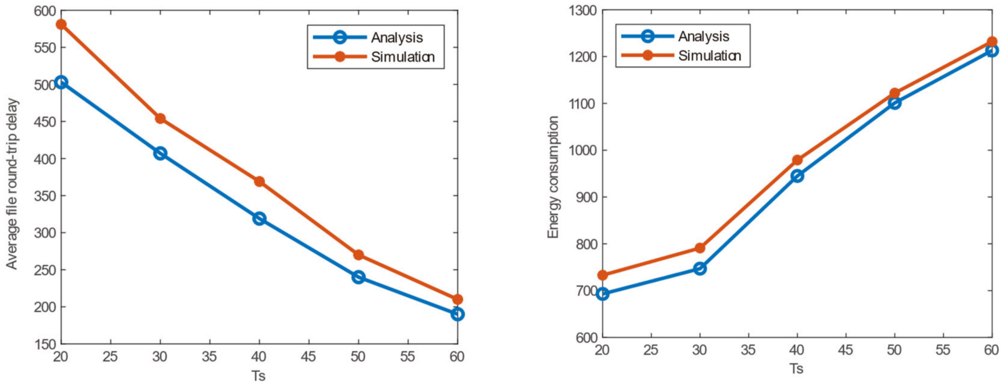
Figures 2–4 show the evolution of the mean file round-trip delay and the corresponding energy consumption against each of the parameters M, T_s, and T_w, both from the analytical model derived in Section 5 and simulations.

Figure 2 presents the mean file round-trip delay and the corresponding energy consumption against M. M is the number of independent coded packets disseminated by the source, which means the redundancy added in the network. We can see that with the increase of M, the file round-trip delay decreased, while the corresponding energy consumption increased. For the coding-based transport scheme in ON, the more coded packets the destination received, the more likely it was to decode the file, and the more energy it would consume to forward those packets. Furthermore, a good match between the analysis and simulation curves could be observed, which validates the accuracy of our model. In addition, it can be seen from the figure that the fit between the analytical model and simulations improved when M increased. For the given T_s and T_w, a lower value of M led to a longer delivery delay, which meant a larger number of cycles, resulting in a bigger inaccuracy in delay and energy consumption.



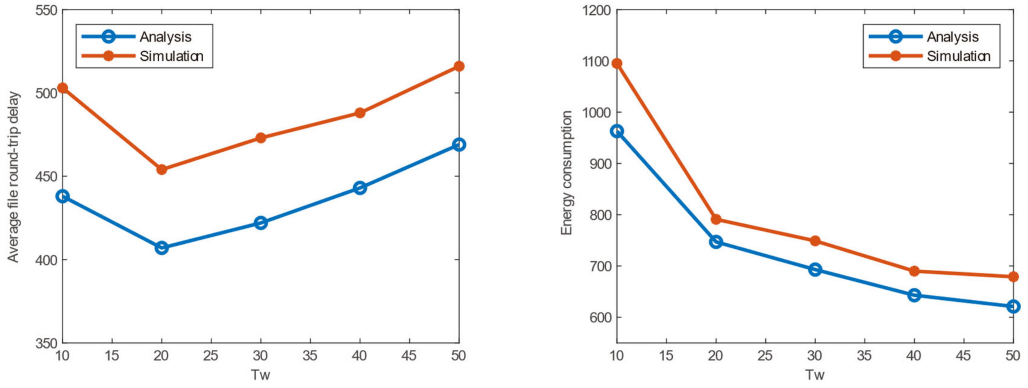
(**a**) (**b**)

Figure 2. $T_s = 30$, $T_w = 20$. (**a**) Mean file round-trip delay against M; (**b**) energy consumption against M.



(**a**) (**b**)

Figure 3. $M = 10$, $T_w = 20$. (**a**) Mean file round-trip delay against T_s; (**b**) energy consumption against T_s.



(**a**) (**b**)

Figure 4. $M = 10$, $T_s = 30$. (**a**) Mean file round-trip delay against T_w; (**b**) energy consumption against T_w.

Figure 3 shows the mean file round-trip delay and the corresponding energy consumption against T_s. T_s is the time in which coded packets are allowed to spread as soon as they have been sent out by the source node. With a larger T_s, the destination can grab the needed number of independent packets more rapidly, which need more forwarding operations. As a result, the file round-trip delay decreased and the corresponding energy consumption increased with the increase of T_s. Similar with Figure 2, a good match between analysis and simulation curves can also be observed in Figure 3a,b, which can validate the accuracy of our model. It also can be seen from the figure that the fit between the analytical model and simulations improved when T_s increased, because for the given M and T_w, a larger T_s could reduce the duration of cycles, and a lower cycle duration would lead to higher accuracy in the theoretical results.

Figure 4 shows the mean file round-trip delay and the corresponding energy consumption against T_w. With the increase of T_w, the file round-trip delay decreased first and then increased; while the energy consumption first dropped rapidly, and then slowed down. This trend can be explained as follows: T_w controlled the time period that the source waited for ACKs instead of controlling the packets' spreading. If T_w was too low, the cycle ended before the latest ACK could reach the source, so more cycles were needed to transfer files, and the delay and energy consumption were large. A moderate increase in T_w could be helpful to alleviate this situation and reduce time delay and energy consumption. But if T_w continued to increase, the network wasted too much time waiting. However, since the number of packet forwarding did not increase, the energy consumption did not increase. This additional waiting time did not cause lots of packets forwarding, so a higher T_w did not lead to more energy consumption. In addition, a good match between analysis and simulation curves can be observed in Figure 4a,b, which can validate the accuracy of our model. Meanwhile, the fit between the analytical model and simulations improved when the delay decreased, as a longer round-trip delay accumulated bigger inaccuracies.

6.2. Performance of the Optimal Procedure

The optimization of the mean file round-trip delay EC_K and the corresponding energy consumption ED_K was related to weight parameter α ($\alpha \in [0, 1)$). In Figures 5–7, we plot the optimal values of the number of coded packets M_r, the spreading time T_{s_r}, the waiting time T_{w_r} ($r \in (1, 2, \ldots, K)$), respectively, for the set of $\alpha = 0,\ 0.25,\ 0.5,\ 0.75$. In Figures 8 and 9, we plot the optimal values of EC_K and ED_K, respectively. In all experiments, we set $TE = 20000$, $DC = 24000$.

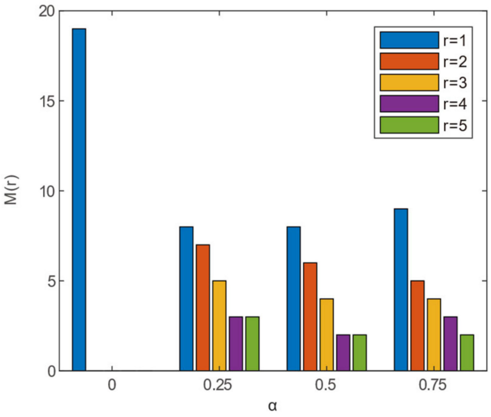


Figure 5. Optimal values of M_r.

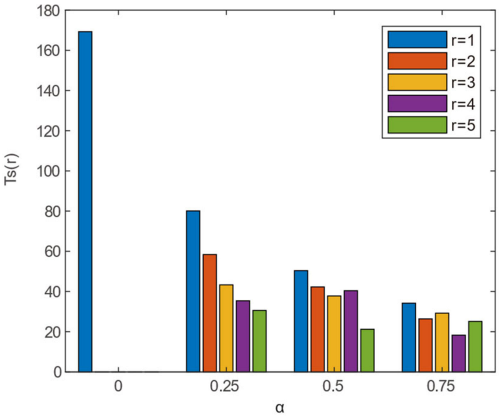


Figure 6. Optimal values of T_{s_r}.

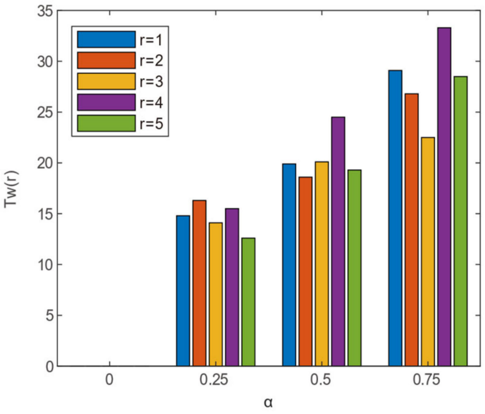


Figure 7. Optimal values of T_{w_r}.

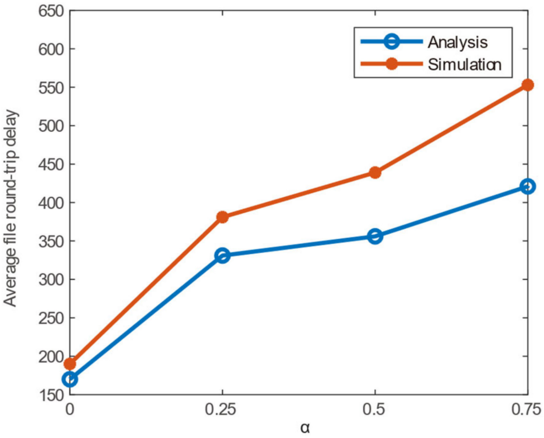


Figure 8. Optimal values of the mean round-trip time EC_K under different α.

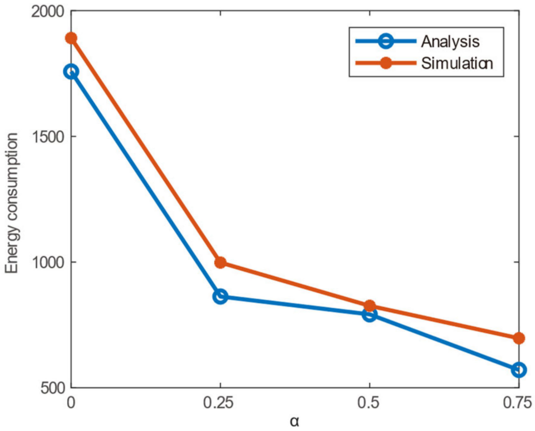


Figure 9. Optimal values of the energy consumption ED_K under different α.

As is shown in Figures 5–7, the optimal values of M_r and T_{s_r} decreased with the increase of the index of a cycle, r, which represented the number of independent packets the destination node still needed to decode the file. Hence, a higher r meant that the source had to add more redundancy (M_r) to the network, and packets needed a longer diffusion time (T_{s_r}) to reach the destination. Note that the maximization of *NDM* was equivalent to the minimization of the round-trip delay ED_K when $\alpha = 0$. Thus, the optimal values of M_r and T_{s_r} in the first cycle, i.e., M_5 and T_{s_5}, were significantly larger than the cases when α got other values. As a result, the delivery of a file accomplished in the first cycle, and the round-trip delay of file was reduced significantly at the cost of energy. The optimal value of

T_{w_r} seemed to be independent of r. Since T_{w_r} controls the time period that the source waits for ACKs in cycle r, it was less affected by r than M_r and T_{s_r}.

In Figures 8 and 9, we compare the optimal values of the EC_K and the ED_K with the simulation results, respectively. It can be observed that the simulation results were in good agreement with the analytical results, under different values of α. This validated our overall optimization procedure based on our model.

6.3. Performance of the Proposed Transport Scheme

The proposed reliable transport scheme (RTS) considered the selfish behavior of relays in practical applications, and made an optimal trade-off between energy consumption and delay. To verify the effectiveness of RTS, the performance was compared with that of a modified version of the RTS (denoted as RTS-MV) under different conditions. In RTS-MV, there was no control or intervention over the redundancy and diffusion process of the coded packets, and file delivery was achieved based on coding and acknowledgement without any optimization in energy consumption. RTS-MV represents a class of methods that considered neither redundancy control nor optimization of the forwarding process. Moreover, the cache management strategy we used is the first-in-first-out (FIFO). We set simulation time TE = 20,000, total energy of relays (the number of packets transmissions allowed) DC = 24,000, and weight parameter $\alpha = 0.5$. The values of $\{M_r, T_{s\_r}, T_{w\_r}\}_{r=1}^{K}$ were set according to the optimal setting provided by the optimal program. The performance metrics we used were the mean round-trip delay of files, the mean energy consumption it took to deliver a file, and the number of files successfully delivered over a certain period.

Note that we did not specify the units of the simulation time in experiments. One can think of simulation time = 20,000 as simulation time = 20,000 s, in which case, $\lambda = 0.01$ has to be $\lambda = 0.01\ \mathrm{s}^{-1}$. Similarly, one can think of simulation time = 20,000 as simulation time = 20,000 min (resp. delay = 20,000 h), in which case, $\lambda = 0.01$ has to be $\lambda = 0.01\ \mathrm{min}^{-1}$ (resp. $\lambda = 0.01\ \mathrm{h}^{-1}$).

Let p denote the level of the residual power that triggers the selfish behavior of node; we compared the performance of the RTS and RTS-MV versus p. In Figures 10–12, the mean round-trip delay of files, the mean energy consumption to deliver a file, and the number of files successfully delivered in the simulation time of both RTS and RTS-MV are illustrated, respectively.

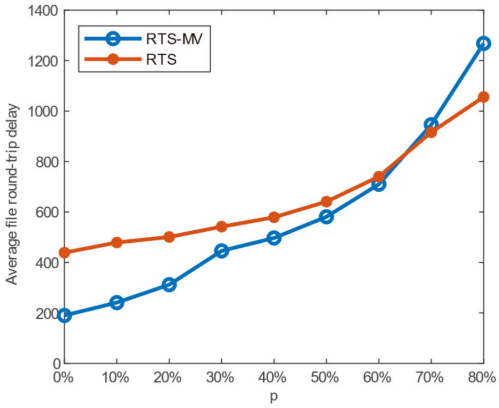


Figure 10. The mean file round-trip delay of both the reliable transport scheme (RTS) and modified version of the RTS (RTS-MV) under different p.

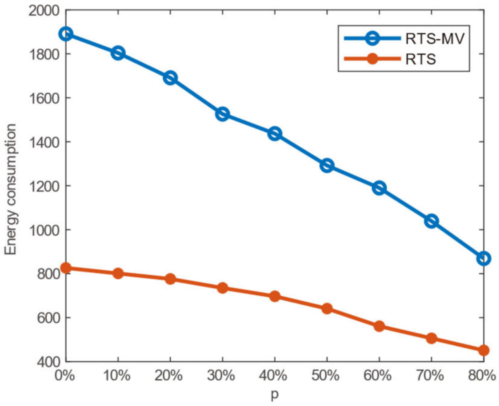


Figure 11. The energy consumption of both RTS and RTS-MV under different *p*.

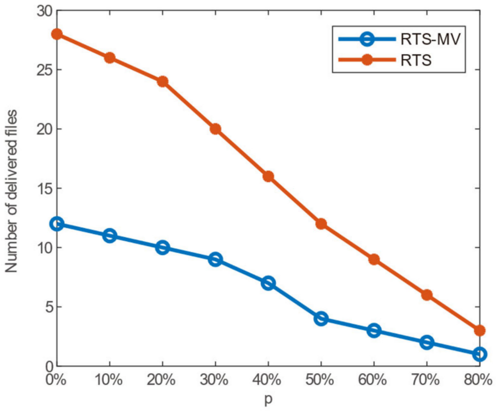


Figure 12. The number of files successfully delivered in the simulation time of both RTS and RTS-MV under different *p*.

It can be seen from the Figures 10–12 that, with the increase of *p*, the mean round-trip delays of both schemes rose greatly, while the energy consumption declined with the increase of *p*, that is, the number of files successfully delivered. A higher *p* meant that the relays' selfish behavior appeared easier, and the number of nodes that were willing to participate in message forwarding became smaller. Hence, the efficiency of file delivery was affected, and the power consumption caused by forwarding/replication was reduced.

What is more, the mean round-trip delay of the RTS was slightly higher than that of the RTS-MV when $p < 70\%$. For example, when $p = 30\%$, RTS had a 26% larger delay than that of RTS-MV. In RTS-MV, to make the destination node decode the file as quickly as possible, the source added a lot of redundancy to the network, and packets were spread and forwarded in the epidemic style for a long time. In other words, RTS-MV got a satisfying delay at the cost of energy. Otherwise, the RTS considered both delay and energy consumption and tried to make a compromise between those two metrics, resulting in a larger delay than that of RTS-MV. However, when *p* was very high, i.e., 70%, the RTS-MV ran out of "free energy" very quickly; then it faced the dilemma that most nodes in the network refused to forward messages, which resulted in a very long delay, and even the failure of the delivery of the file. On the contrary, RTS could extend the time period for relays to participate in the communication service by providing a relatively fair transmission scheme for the source and relays. That is why the mean round-trip delay of the RTS was smaller than that of the RTS-MV when $p \geq 70\%$ (the delay of RTS was reduced by 17% compared to that of RTS-MV when $p = 80\%$).

The mean energy consumption to deliver a file of the RTS was significantly lower than that of the RTS-MV. For instance, when $p = 30\%$, RTS had a 26% lower energy consumption than that of RTS-MV. In order to prevent that the residual power of relays going down to the level that could trigger selfish behavior, the RTS reduced the relay's energy consumption by periodically injecting a small number of coded packets into the network and limiting the time for the dissemination of packets, according to the latest ACK received by the source in the last cycle. However, RTS-MV spent a lot of energy to reduce the delay, and some of the energy was consumed in inefficient replication/forwarding. Therefore, RTS used much less energy consumption than that of RTS-MV.

The number of files successfully delivered in the simulation time of the RTS was remarkably larger than that of the RTS-MV. For example, when $p = 30\%$, RTS had a 26% more delivered files than did RTS-MV. Since it was less likely for relays in the RTS to behave selfishly than in the energy-consuming RTS-MV, RTS could keep high efficiency for a longer time. In addition, the RTS could avoid network congestion by controlling the redundancy injected to the network, which prevented performance degradation caused by heavy consumption of network resources. As a contrast, RTS-MV consumed the limited cache of nodes quickly, resulting in a reduction of transmission efficiency. Therefore, RTS could deliver more files than RTS-MV under the given time and energy constraints.

We also plotted the ratio of nodes with selfish behavior and the number of files delivered successfully versus the simulation time t, shown as Figures 13 and 14, respectively. p was set to 30% for both experiments. It can be observed from Figure 13 that the ratio of nodes with selfish behavior in the RTS-MV went up fast, and then reached saturation. Because the energy of nodes that were willing to provide for forwarding was quickly exhausted to get a low delay, more and more nodes became selfish, as the simulation proceeded. What is more, it also can be seen that the RTS outperformed the RTS-MV most of time. Specifically, the curve showed that the ratio of nodes with selfish behavior rose slowly, and it took much longer to reach its saturation. Based on the analysis, we can conclude that in the RTS, limited energy was efficiently used and the network could provide longer effective communication time compared with the RTS-MV.

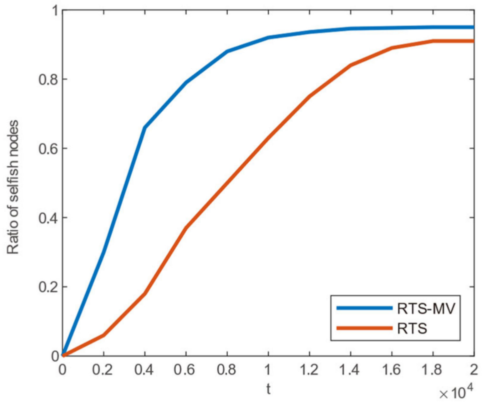


Figure 13. Ratio of nodes with selfish behavior versus simulation time.

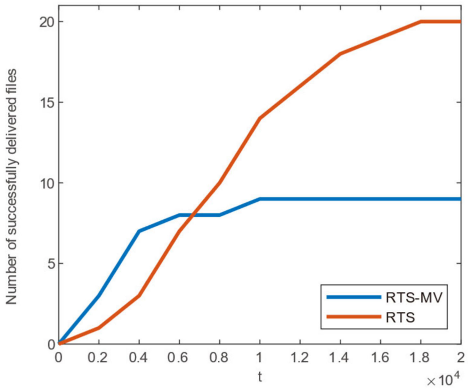


Figure 14. Number of successfully delivered files versus simulation time.

Figure 14 shows the comparison result of the two schemes in terms of the number of successfully delivered files. For the RTS-MV, the number of delivered files grew rapidly to its maximum, and then remained unchanged with no more files delivered successfully. This is because that the network had relatively sufficient energy at the beginning of the simulation, relays were highly active in forwarding, and could deliver files within a short delay by consuming a lot of energy. However, after the energy of relays was exhausted, the number of selfish nodes increased greatly. Since the number of nodes that could participate in communication decreased fast, the delivery probability of files became very low. As for the RTS, the number of files delivered grew slower but more steadily as t increased, and finally the maximum number far exceeded that of the RTS-MV. Because the RTS reduced the energy consumption by limiting the redundancy injected into the network and controlling unnecessary transmissions/replications, the average delay was slightly higher, but the effective communication duration was greatly increased compared to RTS-MV under the same energy constraints. In spite of this, with the proliferation of selfish nodes, the efficiency of the RTS-MV declined dramatically, while the RTS delivered files very steadily and outperformed the RTS-MV. In general, the RTS could use limited energy more efficiently, and was able to keep high efficiency for a longer time.

In addition, the storage congestion caused by load increase also affected the transmission efficiency. In this work, we increased the file size, so that number of pieces *K* that the file is divided into also increased, given that the size individual packet was fixed, and *p* was set to 10%. Then the performances of both RTS and RTS-MV with the change of *K* were investigated. The results are shown as Figures 15–17.

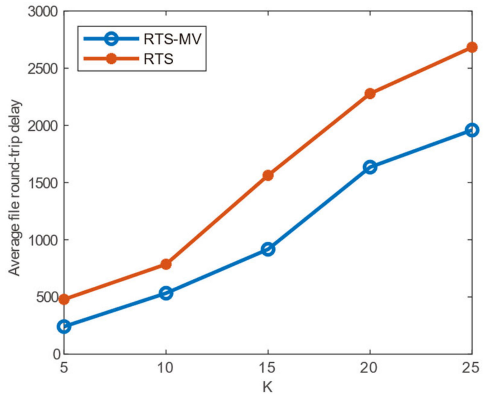


Figure 15. The mean file round-trip delay of both RTS and RTS-MV under different *K*.

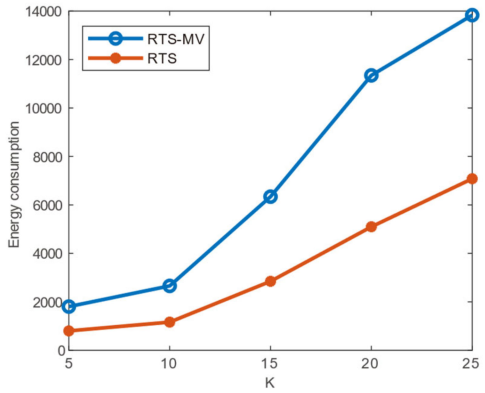


Figure 16. The energy consumption of both RTS and RTS-MV under different *K*.

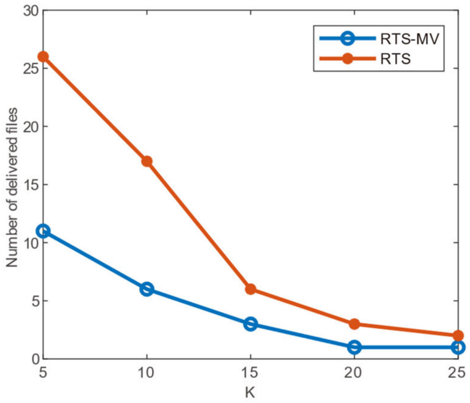


Figure 17. The number of successfully delivered files of both RTS and RTS-MV under different *K*.

It can be seen from the Figures 15–17 that, with the increase of *K*, the mean round-trip delay and energy consumption of both RTS and RTS-MV rose greatly, and the number of files successfully delivered of both RTS and RTS-MV declined significantly. This is because the large load as well as the use of redundancy-based methods (epidemic routing strategy and rateless coding) accelerated the exhaustion of cache, leading to storage congestion. Then the packet-forwarding efficiency was greatly affected. Compared with the RTS-MV, the mean round-trip delay of the RTS was higher, the mean energy consumption to deliver a file of the RTS was significantly lower, while the number of files successfully delivered in the simulation time was remarkably larger. For example, when *K* = 20, RTS had a 38% larger delay than that od RTS-MV, but had a 55% less energy consumption and 67% more delivered files than that of RTS-MV. This is mainly due to the appropriate control of redundancy in the proposed RTS, so as to reduce the consumption of storage and improve the efficiency of message forwarding. These results prove that the proposed transport scheme could somehow achieve congestion control in ONs. In fact, the proposed transport scheme could be combined with an appropriate cache management strategy to achieve more efficient congestion control and more reliable data transfer in human ONs. That will be investigated in our future work.

In summary, the RTS could significantly reduce the energy consumption of file delivery by limiting the redundancy injected into the network and controlling unnecessary transmissions/replications. Although the delivery delay slightly increased, the occurrence of selfish behaviors among nodes could be alleviated, and the number of files successfully delivered could be improved notably. Therefore, the reliability and stability of the communication service in human ONs were enhanced. The RTS was very reliable and suitable for some scenarios where network resources, especially power, were very scarce and the number of messages to be sent was large.

7. Conclusions

In this paper, we proposed a reliable transport scheme, the RTS, for human opportunistic network, where an optimal trade-off between the file round-trip delay and the energy consumption of relays were made. The scheme relied on acknowledgements and coding at the source to enable successful file delivery. The cache management rule referred to as the "full-duplex" strategy was formulated when the network was configured, and a mathematical model was set up to analyze the proposed scheme. This model described the evolution of packets' dissemination and allowed both the mean file round-trip delay and the energy consumption up to the reception of last ACK by the source to be expressed. Based on the model, a new function was proposed to reflect the impact of these two metrics on the number of files delivered under time and energy constraints. Then the function was optimized to obtain the parameters' combination which could achieve optimization of the file delivery delay and the energy consumption. The accuracy of the analytical model was validated through numerous simulations, and results from performance comparison showed that the RTS could significantly reduce the energy consumption of file delivery, which obviously alleviated the selfish behavior of nodes.

Although the delivery delay of the RTS increased slightly compared with that of the energy-consuming approaches, a more reliable and stable communication service in human ONs could be provided. Thus, the proposed RTS was very credible and suitable for some scenarios where network resources, especially power, were very scarce and the number of messages to be sent was quite large.

Future work consists (i) in progressing to a better protocol for human ONs by considering multiple flows, and (ii) in extending this scheme to the more general human ONs with heterogeneous-mobility models.

Author Contributions: M.L., M.T., X.C. and J.W. conceived and designed the proposed scheme. M.L. and M.T. conceived and designed the simulations; M.L. and X.C. performed the simulations; X.C. and J.W. analyzed the data; M.L. wrote the paper. All authors have read and agreed to the published version of the manuscript.

Funding: The research of M.T. and J.W. was supported by the National Natural Science Foundation of China (grant number: U1736121).

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

References

1. Karamshuk, D.; Boldrini, C.; Conti, M.; Passarella, A. Human mobility models for opportunistic networks. *Commun. Mag. IEEE* **2011**, *49*, 157–165. [CrossRef]
2. Raju, B.B.; Chandra, N.S. A study of blockage control design and transmission accuracy in opportunistic networks. *Int. J. Comput. Trends Technol.* **2014**, *14*, 16–23. [CrossRef]
3. Baek, K.M.; Seo, D.Y.; Chung, Y.W. An improved opportunistic routing protocol based on context information of mobile nodes. *Appl. Sci.* **2018**, *8*, 1344. [CrossRef]
4. Aung, C.Y.; Ho, I.W.H.; Chong, P.H.J. Store-carry-cooperative forward routing with information epidemics control for data delivery in opportunistic networks. *IEEE Access* **2017**, *5*, 6608–6625. [CrossRef]
5. Liang, H.; Lars, D. Heterogeneous Community-based mobility model for human opportunistic network. In Proceedings of the IEEE International Conference on Wireless and Mobile Computing Network Communication, Marrakech, Morocco, 12–14 October 2009; pp. 456–470.
6. Humphreys, L. Mobile Social Networks. In *The International Encyclopedia of Digital Communication and Society*; John Wiley Sons Inc.: New York, NY, USA, 2015.
7. Gupta, A.K.; Bhattacharya, I.; Banerjee, P.S.; Mandal, J.K.; Mukherjee, A. Dirmove: Direction of movement based routing in dtn architecture for post-disaster scenario. *Wirel. Netw.* **2016**, *22*, 723–740. [CrossRef]
8. Toor, Y.; Muhlethaler, P.; Laouiti, A.; La Fortelle, A. Vehicle ad hoc networks: Applications and related technical issues. *IEEE Commun. Surv. Tutor.* **2016**, *10*, 74–88. [CrossRef]
9. Bhattacharjee, S.; Roy, S.; Dasbit, S. Dpdrm: A decentralized post-disaster resource management scheme using energy efficient smart phone based dtn. *J. Netw. Comput. Appl.* **2018**, *111*, 1–16. [CrossRef]
10. Liu, M.; Zhang, X.; Xia, S.; Wu, J.; Tian, M. An Efficient Social-Aware Routing Strategy in Mobile Social Networks. Lecture Notes in Computer Science. *Artif. Intell. Secur.* **2019**, *v11633*, 534–545.
11. Wang, H.; Feng, G.; Wang, H.; Lv, H.; Zhou, R. Rabp: Delay/disruption tolerant network routing and buffer management algorithm based on weight. *Int. J. Distrib. Sens. Netw.* **2018**, *14*, 155014771875787. [CrossRef]
12. Wang, R.; Sabbagh, A.; Burleigh, S.C.; Zhao, K.; Qian, Y. Proactive retransmission in delay-/disruption-tolerant networking for reliable deep-space vehicle communications. *IEEE Trans. Veh. Technol.* **2018**, *67*, 9983–9994. [CrossRef]
13. Yang, G.; Wang, R.; Zhao, K.; Zhang, X.; Li, W.; He, X. Queueing analysis of dtn protocols in deep-space communications. *IEEE Aerosp. Electron. Syst. Mag.* **2019**, *33*, 40–48. [CrossRef]
14. Luo, Q.; Wang, J.; Liu, S. Aeromrp: A multipath reliable transport protocol for aeronautical ad hoc networks. *IEEE Internet Things J.* **2018**, *6*, 3399–3410. [CrossRef]
15. Marek, A.; Jaroslaw, K. On applicability of network coding technique for 6lowpan-based sensor networks. *Sensors* **2018**, *18*, 1718.
16. Tsiory, R.A.; Tsuru, M. When Does Network Coding Benefit Store-Carry-and-Forwarding Networks in Practice. In *International Conference on Intelligent Networking and Collaborative Systems*; Springer: Cham, Switzerland, 2018.

17. Shnaiwer, Y.N.; Sorour, S.; Al-Naffouri, T.Y.; Al-Ghadhban, S.N. Opportunistic network coding-assisted cloud offloading in heterogeneous fog radio access networks. *IEEE Access* **2019**, *7*, 56147–56162. [CrossRef]
18. Ding, S.; He, X.; Wang, J.; Liu, J. Pre-decoding recovery mechanism for network coding opportunistic routing in delay tolerant networks. *IEEE Access* **2018**, *6*, 14130–14140. [CrossRef]
19. Hu, G.; Xu, K.; Xu, Y. Deadline-aware opportunistic network coding for multi-relay-aided single-source single-destination network. *IEEE Commun. Lett.* **2017**, *21*, 2282–2285. [CrossRef]
20. Zeng, D.; Guo, S.; Hu, J. Reliable bulk-data dissemination in delay tolerant networks. *IEEE Trans. Parallel Distrib. Syst.* **2014**, *25*, 2180–2189. [CrossRef]
21. Douik, A.; Sorour, S.; Tembine, H.; Al-Naffouri, T.Y.; Alouini, M.S. A game-theoretic framework for network coding based device-to-device communications. *IEEE Trans. Mob. Comput.* **2017**, *16*, 901–917. [CrossRef]
22. Parisis, G.; Sourlas, V.; Katsaros, K.V.; Chai, W.K.; Pavlou, G.; Wakeman, I. Efficient content delivery through fountain coding in opportunistic information-centric networks. *Comput. Commun.* **2017**, *100*, 118–128. [CrossRef]
23. Ali, A.; Altman, E.; Chahed, T.; Panda, M.K.; Sassatelli, L. A new reliable transport scheme in Delay Tolerant Networks based on acknowledgments and random linear coding. In Proceedings of the 2011 23rd International Teletraffic Congress (ITC), San Francisco, CA, USA, 6–9 September 2011.
24. Sassatelli, L.; Ali, A.; Panda, M.; Chahed, T.; Altman, E. Reliable transport in delay-tolerant networks with opportunistic routing. *Wirel. Commun. IEEE Trans.* **2014**, *13*, 5546–5557. [CrossRef]
25. Park, M.; Paek, J. Taim: Tcp assistant-in-the-middle for multihop low-power and lossy networks in iot. *J. Commun. Netw.* **2019**, *21*, 192–199. [CrossRef]
26. Ummto, S.H.; Algeria, U.; Lloret, J.; Lorenz, P. Tcp performance in mobile ad hoc networks. In *Network Protocols Algorithms*; Macrothink Institute: Vegas, NV, USA, 2014.
27. Ahn, G.S.; Campbell, A.T.; Veres, A.; Sun, L.H. Supporting service differentiation for real-time and best-effort traffic in stateless wireless ad hoc networks (swan). *IEEE Trans. Mob. Comput.* **2002**, *1*, 192–207.
28. Liu, S. Singh. ATCP: TCP for mobile Ad Hoc networks. *IEEE J. Sel. Areas Commun.* **2001**, *19*, 1300–1315. [CrossRef]
29. Alessi, N.; Burleigh, S.; Caini, C.; Cola, T.D. Design and performance evaluation of ltp enhancements for lossy space channels. *Int. J. Satell. Commun. Netw.* **2019**, *37*, 3–14. [CrossRef]
30. Zafar, A.; Shahriar, M.; Atiquzzaman, M.; Ivancic, W.D.; Wood, L. A sender-based TFRC for Saratoga: A rate control mechanism for a space-friendly transfer protocol. In Proceedings of the 2011 Aerospace Conference, Big Sky, MT, USA, 5–12 March 2011.
31. Scott, K.; Burleigh, S. Bundle Protocol Specification. Internet Engineering Task Force Request For Comments (IETF RFC). 2007. Available online: https://www.hjp.at/doc/rfc/rfc5050.html (accessed on 25 August 2020).
32. Whisman, D.; Médard, M.; Kusuma, J.; Croux, A. Benefits of Opportunistic Routing, Implicit Acknowledgments, and Network Coding on a Linear Broadcast Network. In Proceedings of the 2016 IEEE Globecom Workshops (GC Wkshps), Washington, DC, USA, 4–8 December 2016.
33. Zeng, D.; Guo, S.; Jin, H.; Leung, V. Dynamic segmented network coding for reliable data dissemination in delay tolerant networks. In Proceedings of the IEEE International Conference on Communications, Ottawa, ON, Canada, 10–15 June 2012; IEEE: New York, NY, USA, 2012.
34. Thai, T.T.; Chaganti, V.G.; Lochin, E.; Lacan, J.; Dubois, E.; Gelard, P. Enabling E2E reliable communications with adaptive re-encoding over Delay Tolerant Networks. In Proceedings of the 2015 IEEE International Conference on Communications (ICC), London, UK, 8–12 June 2015.
35. Carlos, B.; Joan, B.; Sergi, R. Efficient broadcast in opportunistic networks using optimal stopping theory. *Ad Hoc Netw.* **2019**, *88*, 5–17.
36. Spyropoulos, T.; Psounis, K.; Raghavendra, C.S. Performance analysis of mobility-assisted routing. In Proceedings of the 7th ACM International Symposium on Mobile Ad Hoc Networking and Computing, Florence, Italy, 22–25 May 2006.
37. Groenevelt, R.; Nain, P.; Koole, G. Message delay in manet. *ACM Sigmetrics Perform. Eval. Rev.* **2005**, *33*, 412. [CrossRef]
38. Vahdat, A.; Becker, D. Epidemic routing for partially-connected ad hoc networks. In *Handbook of Systemic Autoimmune Diseases*; Elsevier: Amsterdam, The Netherlands, 2000.
39. Yan, Z.; Jianhua, C.; Meng, T. Distributed it codes on multiple layers networks. *J. Electron. Inf. Technol.* **2019**, *2*, 293–307.

40. Rao, K.D. LT and Raptor Codes. In *Channel Coding Techniques for Wireless Communications*; Springer: New Delhi, India, 2015.
41. Bouras, C.; Kanakis, N.; Kokkinos, V.; Papazois, A. Enhancing Reliable Mobile Multicasting with RaptorQ FEC. In Proceedings of the 2012 IEEE Symposium on Computers and Communications (ISCC), Cappadocia, Turkey, 1–4 July 2012.
42. Martin-Campillo, A.; Marti, R. Energy-efficient forwarding mechanism for wireless opportunistic networks in emergency scenarios. *Comput. Commun.* **2012**, *35*, 1715–1724. [CrossRef]
43. Storn, R.; Price, K. Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces. *J. Glob. Optim.* **1997**, *11*, 341–359. [CrossRef]