Impact of Selective Dropping Attacks on Network Coding Performance in DTNs and a Potential Mitigation Scheme

M. Chuah, P. Yang
Department of Computer Science & Engineering
Lehigh University
chuah@cse.lehigh.edu, pey204@lehigh.edu

***Abstract*—** Some ad hoc network scenarios are characterized by frequent partitions and intermittent connectivity. A store-and-forward network architecture known as the disruption tolerant network (DTN) has been designed for such challenging network environments. To further improve the delivery performance, some researchers have proposed some network coding schemes for DTNs. However, not much papers discuss the security issues of network coding schemes in DTNs. In this paper, we first discuss some attacks that can be launched against network coding schemes in DTNs. Then, we focus on evaluating the impact of selective data dropping attacks on the delivery performance of a network coding scheme we design for DTN. Next, we describe a mitigation scheme that we design to overcome such attacks. Our mitigation scheme uses dynamic redundancy factor to generate more coded packets when a source notices performance degradation in the delivery performance. Via simulation studies, we show that our mitigation scheme is effective in restoring the performance degradation caused by the selective dropping attacks as long as alternate DTN paths exist for a source/destination pair.

***Index Terms*—disruption tolerant networking, network coding, security, mitigation scheme, redundancy.**

I. INTRODUCTION

With the advancement in technology, many users carry computing devices e.g. PDAs, cell-phones etc with wireless interfaces. Such devices can form mobile ad hoc networks and communicate with one another via the help of intermediate nodes. Such ad hoc networks are very useful in several scenarios e.g. battlefield operations, vehicular ad hoc networks and disaster response scenarios. Many ad hoc routing schemes have been designed for ad hoc networks but such routing schemes are not useful in some challenging network scenarios where the nodes have intermittent connectivity and suffer frequent partitioning. Recently, disruption tolerant network technologies [1],[2] have been proposed to allow nodes in such extreme networking environment to communicate with one another. Several DTN routing schemes [3],[4],[5],[6] have been proposed. In addition, some researchers have proposed using randomized network coding [11],[12] technique to enhance the delivery performance of a batch of messages in DTNs. These existing schemes are designed assuming the nodes move according to a homogeneous mobility model and hence may not work well if the nodes move according to a non-homogeneous mobility model. Thus, in [23], we design a network-coding scheme that will work better when the nodes move according to a non-homogeneous mobility model. Our scheme allows coded packets to be distributed to more popular nodes and hence can achieve higher message delivery ratio than those schemes that are designed assuming a homogeneous mobility model. We conduct extensive simulation studies for unicast communications to demonstrate the superiority of our designed scheme. However, we did not address any security issues in [23]. In this paper, we first discuss various attacks that can be launched in DTNs and discuss how the attacks can affect the delivery performance of network coding schemes in DTNs. Then, we focus on one particular type of attacks, namely the selective dropping data attack, and describe a mitigation scheme that can be used to deal with such attacks. Via simulation studies, we show that our scheme is quite effective in dealing with selective data dropping attacks as long as alternate DTN paths exist between any source/destination pair.

The remainder of this paper is organized as follows. We provide a brief review of related work in Section II. We describe various network coding schemes that are proposed for DTNs. In Section III, we discuss various security issues in network coding enabled DTNs. In Section IV, we describe a mitigation scheme that deals with selective data dropping attack that we design. In Section V, we describe the simulation model we use to evaluate our mitigation scheme. We also present simulation results that show the effectiveness of our designed mitigation scheme in dealing with selective dropping data attacks on both homogeneous and non-homogeneous mobility models. We conclude in Section VI some discussions on future work.

II. RELATED WORK

A. Network Coding Schemes for DTNs

Previous studies have proposed to use erasure coding to deal with network disruptions in DTNs [7],[8]. It has been shown that network coding [9] can improve the throughput in wireless communication. However, in DTNs, a node seldom has more than one neighbor, and such wireless

coding opportunities rarely occur. In [11], the authors propose a scheme called the network coding based epidemic routing (NCER) scheme which transmits a batch of data packets with network coding. In this scheme, when two nodes meet, they transmit coded packets to each other. A coded packet x is a linear combination of K source packets, E_1 … E_K in the form $x = \sum_{i=1}^{K} \alpha_i E_i$ where α_i is the coding coefficient. Suppose that node *a* holds *m* coded packets in its buffer, node a encodes all coded packets in its buffer, namely x_1 .. x_m to generate a coded packet x_a :

$$x_a = \sum_{i=1}^{m} \beta_i x_i$$

where all multiplication and addition operations are defined on a Galois Filed and is randomly chosen from the field. It is easy to see that x_a is also a linear combination of the K original packets, and the coefficients can be derived. Node *a* then transmits x_a along with its coding coefficients over the original packets to node *b*. When node *b* receives x_a , it stores x_a in its buffer if space is available. Otherwise, node *b* encodes x_a with each packet in its buffer as follows: $x_i^{'} = x_i^{'} + \gamma_i x_a$, where $x_i^{'}$ represents the i^{th} coded packet in the buffer of node *b*, and γ_i is randomly chosen from the Galois Field.

The destination obtains a coded packet when it meets another node, and attempts the decoding process to retrieve K source packets as long as K coded packets have been collected. Because the coding coefficients and the coded packets are known, each coded packet represents a linear equation with the K source packets as unknown variables. Decoding the K source packets is equivalent to solving the linear system composed of K coded packets. In NCER, the nodes keep exchanging the packets until they receive an ACK from the destination that all K packets have been received or the TTLs of the packets have expired. Thus, this scheme is not quite efficient.

In [12], the authors propose another scheme called the efficient network coding protocol (E-NCP). In this scheme, the source transmits slightly more than K coded packets such that these coded packets are sufficient to decode the original packets with high probability. All these coded packets are referred to as pseudo source packets. Each pseudo source packet is then disseminated to L random nodes in the network in the same spirit as the binary spraying scheme in [10]. The authors in [10] have shown that binary spraying is the optimal spraying method with the minimal packet transmission delay under a homogenous mobility model. By adjusting L, one can tune the trade off between the number of relay transmissions and the packet transmission delay. Each relay node keeps a list of tuples: *<i,l>* where *i* and *l* denote the index of a pseudo source packet and its associated spray counter. When node *a* meets node *b,* they exchange their lists. Then, node *a* checks the spray lists in both nodes. If node *a* finds in its own list any tuple that node b does not have, then node *a* transmits a coded packet to node *b*; otherwise node *a* skips the transmission opportunity. If node *a* decides to transmit, it generates a coded packet which is a random linear combinations of all coded packets in its buffer and sets the packet index *i* and the new spray counter *l/2* to the newly generated coded packet. Node *a* then updates its tuple with *<i, l/2>*. Upon receiving a coded packet, node *b* stores or encodes the coded packet and inserts a new tuple into its list:*<i, l>* where *i* and *l* are the index and spray counter carried in the incoming coded packet.

In [23], we describe a network coding scheme called the CANCO scheme which we design for scenarios where the nodes move using non-homogeneous mobility models. In the CANCO scheme, we use two metrics, namely the friendliness metric and the delivery predictability. The friendliness metric measures how popular a node is while the delivery predictability estimates the probability of reaching another node. The friendliness metric helps to distinguish between globally and locally moving nodes. The delivery predictability metric is used to distinguish nodes that are closer to the destination. In our CANCO scheme, the delivery predictability metric of a node to the destination is computed in the same manner as in the Prophet [3] routing scheme. Each node periodically sends a beacon that contains its delivery predictabilities to all other nodes in the network. Each node that hears another node's beacon updates its own delivery predictabilities according to the following three equations:

$$P(a,b) = P(a,b)_{old} + (1 - P(a,b)_{old}) * \alpha \qquad \text{Eq(1a)}$$

$$P(a,b) = P(a,b)_{old} \times \gamma^{k} \qquad \text{Eq(1b)}$$

$$P(a,c) = P(a,c)_{old} + (1 - P(a,c)_{old}) * P(a,b) * P(b,c) * \beta \qquad \text{Eq(1c)}$$

In [3], α is set to 0.75, β is set to 0.25 and γ is set to 0.98.

Each message is divided into K packets and M*K coded packets are generated by a source node. Only K coded packets are needed by a receiver to reconstruct the original message. The source node binary spreads the M*K coded packets to the nodes it chooses. Unlike the MORE protocol [14] where the source node can only transmit a new message after receiving an acknowledgement from the receiver, our CANCO scheme allows a source node to send coded packets from another new message once it has completed its transmissions of all the M*K coded packets of a message. An intermediate node uses the friendliness metric to decide if it wants to spread coded packets to another node it encounters. To avoid spreading the coded packets to too many intermediate node, a minimum batch Bmin value can be set such that a source or intermediate node stops spreading coded packets when it only has Bmin or less coded packets. The pseudo code for the CANCO scheme is shown in Figure 1.

Unicast Communication
Each node, n, maintains a vector, Pdel, which contains delivery predictabilities to other nodes in the network. Let $p_i(d)$ be the delivery predictability of node i to the destination d. For every message j, we denote dest(j) as the destination of message j and packets(j) as all coded packets that are created from message j.

When node n meets another node m, both nodes exchange their delivery predictability vectors. Then, node n executes the following algorithm:

```
For every queued message j,
   if ((p_n(dest(j)) < p_m(dest(j))) && (friendliness(m) > Th1))
     Binary spread packets(j) to node m
   else
     Do nothing.
End For Loop
```

Figure 1: Pseudo Code for the CANCO scheme

III. Security Issues in Network Coding Enabled Wireless Networks

Selfish or malicious nodes may issue false information in either the control or data planes of a network coding-enabled wireless network. When launched in the control plane, such attacks affect how routes are selected. When launched in the data plane, the attacks can corrupt coded packets so that a receiver cannot reconstruct a message.

Control Plane Attacks

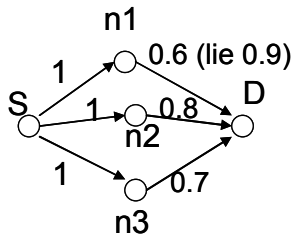


Fig 2: False Link Quality Attacks

a) False Information Regarding Link Quality & Neighbor Set [20,21]

In coding protocols like COPE [9], each node periodically reports its packet reception to its neighbors via local broadcasts. Link qualities are also used to estimate delivery probability. A malicious node can lie about the link qualities it observes and hence can affect how a particular route will be chosen. For example, in Fig. 2, if every node is honest about its link qualities, then D will pick the route S-n_2-D and informs S. However, if n_1 lies that the link quality for the link n_1-D is 0.9, then, D will pick the route S-n_1-D instead. In coding protocols that use global information (e.g., DCAR [19]), each node keeps track of all other nodes in the network that can overhear a packet by maintaining the set of all neighboring nodes on the path of the packet. An attacker can cause incorrect collection of neighboring set information by modifying route request packets or by using worm holes to introduce fake links. Existing approaches for securing source routing protocols, e.g., Ariadne [15], can be used to authenticate the neighboring set information and prevent malicious modifications. However, such techniques cannot deal with insider attacks, i.e., when compromised nodes, which can participate correctly in secure routing procedure, launch data plane attacks.

b) False Topology Information via Wormhole Attacks[21]: Malicious nodes can also introduce fictitious links between honest nodes. As a result, they corrupt nodes' perception of network topology. Existing wormhole solutions, e.g., Packet Leases [16], TrueLink [17], connectivity based solution [18], can be used, but they typically incur huge overhead and thus may nullify the performance gain of network coding.

Data Plane Attacks

a) Corrupted Coded Packets: In such attacks, the attacking node sends corrupted packets into the network. Since each forwarding node combines received packets to form new coded packets, such attacks can have an epidemic effect, as polluted coded packets affect other honest nodes.

b) Selective Packet Dropping [20,21,22]

In current network coding schemes, forwarding nodes are selected such that the total number of transmissions is minimized. As a result, if a malicious node within the routing path drops some or all coded packets, it can significantly degrade the message delivery success rate.

c) False ACK Packet Injections/Dropping of ACK Packets [21]

An attacking node can create bogus ACK packets and prompt the source to prematurely start sending the next batch of coded packets. The destination in turn may receive only partial batches and may not be able to decode any data packets. Such attacks can be mitigated with message authentication but compromised nodes can generate ACK packets with false information that can be correctly authenticated. A malicious node can also report incorrect packet reception information to their neighbors, resulting in coded packets that cannot be decoded by intended next hop nodes. These undecodable packets cannot be acknowledged, and hence the sending node will be forced to repeatedly send packets and ultimately waste bandwidth. Alternatively, if a malicious node drops ACKs, the source will continue transmitting a batch of coded packets several times before giving up. Typically, a malicious node will launch such an attack after it successfully manipulates route metrics to enhance its chances of being selected on the ACK delivery path.

IV. Mitigation Scheme Against Selective Dropping Attacks

In the previous section, we discuss different types of attacks. In the remaining sections, we focus on the selective data dropping attacks. When such an attack is launched, each receiving node may not acquire sufficient number of coded packets to reconstruct messages. Next, we describe a scheme we design to mitigate against such an attack. Our mitigation scheme requires a destination node to measure the delivery ratio for a flow and send the estimated delivery

ratio to the sending node. When a sending node receives the feedback from its receiver, the sending node will adjust the redundancy factor dynamically to mitigate against the degradation in the delivery ratio caused by the attack. Let us assume that the average delay ratio observed by the receiver is DR_{normal} and the redundancy factor used is RF_{normal}. The redundancy factor used by the sending node, $RF_{current}$ is computed as follows: $RF_{current} = \frac{DR_{normal}}{DR_{current}} * RF_{normal}$ where $DR_{current}$ is the currently observed delivery ratio sent by the receiving node. In our simulation studies, the observation window for the delivery ratio is set to 100s.

V. SIMULATION STUDIES

A. Simulation Setup

Mobility Model

By default, the nodes move according to the community-based (CB) mobility model [10]. In the CB model, 50% of the nodes move locally (within 1% of the whole area) and 50% of the nodes move globally (within the whole area). Each locally moving node randomly selects a destination location within its local area and move towards it using a constant speed v. Once it reaches the destination, it repeats its action (i.e. picks another destination to move to). The globally moving nodes behave similarly except that they are allowed to move within the whole area. In our simulation, the speed of all nodes is chosen randomly between (1,5) m/s.

Traffic Model

For the traffic model, we use 4 flows, the source and destination of each flow is randomly chosen. We let the source generates traffic after 15000s and stops the traffic after 35000s. The simulation will continue to run for another Texp time. The default value for Texp is 20,000s. Each reported data point is the average of 5 or 10 runs. By default, each message is divided into 1000 packets and 2000 coded packets are generated from these 1000 packets. A receiver only needs to receive 1000 coded packets to be able to reconstruct the original message.

Attack Model

Each attacking node will drop all the coded packets it receives. We can configure the number of attacking nodes in each simulation run. Each attacking node is randomly selected from those nodes who are neither a source or a destination node.

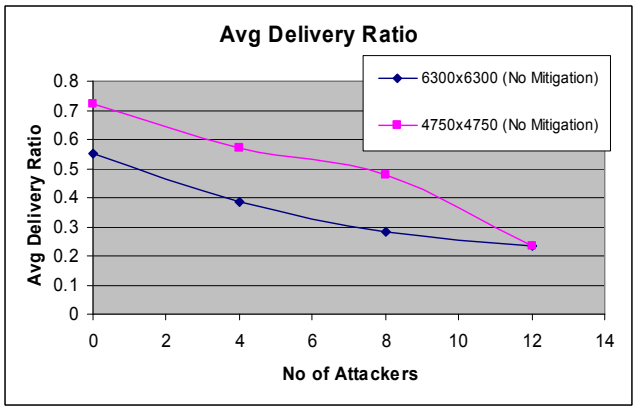
Performance Metrics

The performance metrics we used in our evaluation are: (1) *Delivery Ratio*, which is the ratio of the number of successful messages each destination receives vs the number of messages sent by each sender.

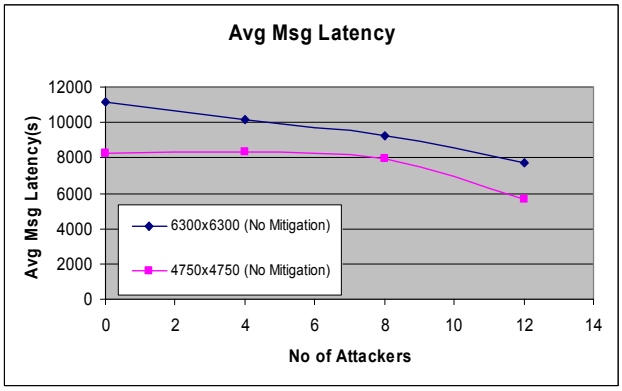
(2) *Average Delay,* which is defined as the average end-to-end delay incurred by the delivered messages, and
(3) Data Efficiency, which is the total number of messages received divided by the number of transmissions used to deliver such messages.

B. Impact of the Selective Dropping Attacks

In this section, we first evaluate the impact of selective dropping attacks on two network scenarios, namely (i) 100 nodes distributed over 4750x4750m^2(referred to as Scenario 1) and (ii) 100 nodes distributed over 6300x6300m^2 (referred to as Scenario 2). In both scenarios, the nodes move according to the CB model. The CANCO scheme is used to distribute the coded packets. There are 4 unicast flows in the network. The message rate is set at 1 msg/200s. A source or an intermediate node stops disseminating coded packets when the number of coded packets it has is equal or below Bmin= 100 coded packets. The message expiration time is set to be 5.5 hour. Figures 3(a) and 3(b) plots the delivery ratio and the average message latency observed when 0, 4, 8, and 12 attacking nodes are present in the system. From the plot in Fig 3(a), one can see that the delivery ratio drops by 68% (compared to no attack case) for Scenario 1 and drops by 57.8% (compared to the no attack case) for Scenario 2 when there are 12 attackers in the network. As the number of attackers increases, only those messages that can be easily delivered become successful and hence the average message latency drops (see Fig 3(b)). Note that the average message latencies are only computed over successfully delivered messages.



(a) Average delivery ratio vs No of Attackers



(b) Avg Msg Latency vs No of Attackers

Figure 3: Impact of Selective Dropping Data Attack on the Delivery Performance of Network Coded Packets

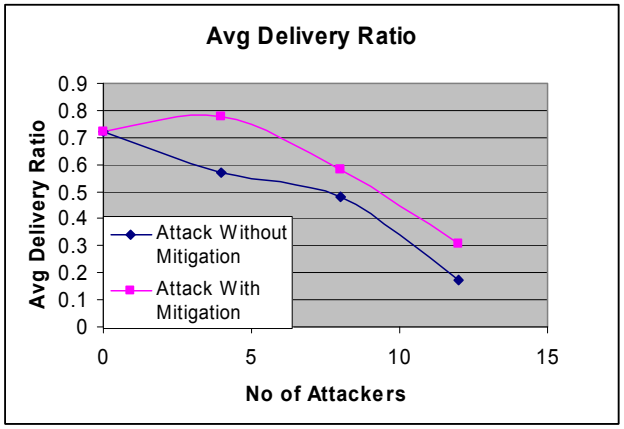
C. Effectiveness of our mitigation scheme

In this subsection, we use the network scenario with 100 nodes distributed over an area of 4750x4750 m^2. We simulate scenarios without attack as well as with 4, 8 and

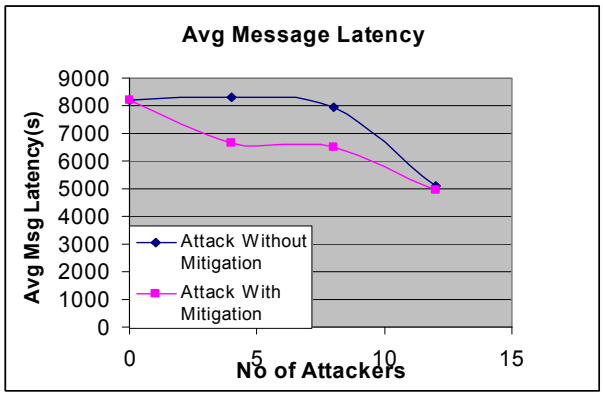
12 attacking nodes. Each attacking node drops 100% of the coded packets that they receive. We simulate the cases with and without the mitigation scheme being turned on. Figures 4(a), (b), (c) show the average delivery ratio, average message latency and the data efficiency we obtained for the case without and with the mitigation scheme as the number of malicious attackers is varied from 0 to 4, 8, and 12. From Fig 4(a), we see that the mitigation scheme was able to improve the delivery ratio by 17-43%. With 4 attackers, the delivery ratio without the mitigation scheme is 0.569 but with mitigation scheme, it is 77.9%. Without attack, the delivery ratio is only 72.3%. This is because a fixed redundancy factor of 2 is used for the no attack case but this value is dynamic for the attacking case. If a dynamic redundancy factor is used in the non-attack case, then the achieved delivery ratio is also close to 77.9%. When there are more attackers, then only messages that can be easily delivered can be successful which explained why the average message latency drops with an increasing number of attackers. The average data efficiency for the case with the mitigation scheme is lower since a higher redundancy factor is used to generate more coded packets to mitigate against the data dropping attacks launched by the attackers.

D. Impact of Mobility Models

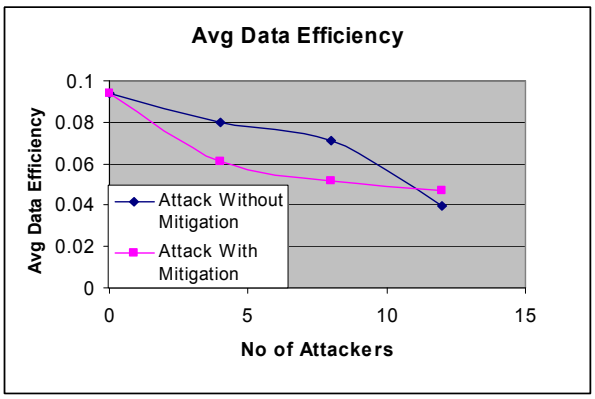
The above results show that our mitigation scheme is quite effective in reducing the impact of the selective dropping attacks in scenarios with non-homogeneous mobility models. We are interested in exploring if it is equally effective in scenarios with homogenous mobility models. We use a network scenario with 100 nodes distributed over 4750x4750m^2 but the nodes move according to the random waypoint model. Since the inter-node encounter time for the RWP model is shorter than the CB model used in previous sections, we set Texp=3 hr instead of the 5.5 hr used for the CB model. We use 12 attackers since the impact of having only 4 attackers on delivery ratio is minimal in the RWP model for all 100 nodes can be used as relaying nodes, and hence there are more alternative paths than the CB model.



(a) Avg Delivery Ratio



(b) Avg Msg Latency



(c) Average Data Efficiency

Figure 4: Effectiveness of Mitigation Scheme on the Delivery Performance of Network Coded Packets.

We tabulate our results in Table 1. The first row shows the results obtained for the average delivery ratio (DE), average message latency (Avg Dly), the average data efficiency (DE) for the case with no attack. The 2nd row shows the results with 12 attackers but without the mitigation scheme while the third row shows the results when the mitigation scheme is turned on. One can observe that the delivery ratio drops by 17% with 12 malicious nodes launching selective data dropping attacks. When the mitigation scheme is turned on, the achieved delivery ratio is 98% of what is observed without the attack. Note that the data efficiency drops when the mitigation scheme is turned on since more redundant coded packets are generated, and relayed by intermediate nodes. For easy comparison, we also list the results we obtained in the earlier section for the CB model with 12 attackers and Texp=5.5 hr in Table 2. Without attack, we see that the delivery ratio for the CB model with Texp=5.5 hr can only be 0.723. This is due to the extremely long inter-node meeting time since the source and destination nodes are chosen from the pool of locally-moving nodes. The impact of 12 attackers on the delivery ratio for the CB model is much greater. The delivery ratio drops by 68% without the mitigation scheme. Even with the mitigation scheme, one can only recover 63.6% of the original delivery ratio. This is due to the fact that for one particular flow, besides the attacking nodes, there is no alternative candidate node that can be used as a relaying node for coded packets. So, even if more coded packets are being generated, they are still being dropped by the

attacking nodes. We can add in a feature to identify malicious nodes and let intermediate nodes avoid such nodes. This will allow us to achieve higher delivery ratio at the expense of increasing the average message latency. We intend to evaluate the impact of having this feature in the near future.

No of MN	DE	Avg Dly	DE
0	0.9065	4631.7	0.1062
12 (No Mitigation)	0.7515	4548	0.1059
12 (With Mitigation)	0.8875	3466.9	0.0324

Table 1: Simulation Results for the RWP Model.

No of MN	DE	Dly	DE
0	0.723	8220.3	0.094
12 (No Mitigation)	0.232	5630.5	0.048
12 (With Mitigation)	0.46	5787.9	0.057

Table 2: Simulation Results for the CB Model.

VI. CONCLUDING REMARKS

In this paper, we have presented the various attacks that can be launched in a DTN that utilizes the network coding feature. We also describe a mitigation scheme that can be used to deal with selective dropping attacks in a DTN that uses the network coding feature. Our simulation results show that our mitigation scheme is effective in improving the delivery ratio as long as there are alternative paths that can bypass the attacking nodes.

In this paper, we merely study the impact of selective dropping attacks on the delivery performance of network coding schemes using the CB and RWP models. We intend to explore the impact of such attacks using more mobility models. In addition, we intend to develop an analytical framework to estimate the delivery performance in the presence of selective dropping attacks. Furthermore, we also intend to study the impacts of other attacks and design mitigation schemes against such attacks.

ACKNOWLEDGMENT

This work has been supported by DARPA under Contract W15P7T-06-C-P430. Any opinions, findings, and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the sponsor of this work.

REFERENCES

[1] K. Fall, " A delay tolerant network architecture for challengednetworks", Proceedings of ACM Sigcomm, 2003..

[2] V. Cerf et al, "Delay Tolerant Networking Architecture", RFC4838, April, 2007

[3] A. Lingren et al, "Probabilistic Routing in Intermittently Connected Networks", Proceedings of Workshop on Service Assurance with Partial and Intermittent Resources, Aug, 2004.

[4] J. Burgess et al, "MaxProp: Routing for vehicle-based disruption tolerant networks", Proceedings of IEEE Infocom, 2006.

[5] M.M.B.Tariq, M. Ammar, and E. Zegura, "Message Ferry Route Design for Sparse Ad hoc Networks with Mobile Nodes", ACM MobiHoc , May22-25, 2006.

[6] S.Jain, K.Fall, and R. Patra, "Routing in a Delay Tolerant Network", SIGCOMM'04, Aug. 30-Sept. 3, 2004.

[7] Y. Wang et al, "Erasure-Coding Based Routing for Opportunistic Networks", Proceedings of ACM workshop on WDTN, 2005.

[8] S. Jain, M. Demmer, R. Patra, K. Fall, "Using Redundancy to Cope with Failures in a Delay Tolerant Network", Proceedings of ACM Sigcomm, Philadelphia,PA, Aug, 2005.

[9] S. Katti et al, "XORs in the Air: Practical Wireless Network Coding", Proceedings of ACM Sigcomm, 2006.

[10] T. Spyropoulos et al, "Efficient routing in intermittently connected mobile networks: multiple copy case" to appear in IEEE/ACM Transactions on Networking, 2007.

[11] Y. Lin, B. Liang, B. Li, "Performance Modeling of Network Coding in Epidemic Routing", Proceedings of ACM MobiOpp, 2007.

[12] Y. Lin, B. Li, B. Liang, "Efficient Network Coded Data Transmissions In Disruption Tolerant Networks", Proceedings of IEEE Infocom, April, 2008.

[13] "The network simulator ns-2", [Online] at http://www.isi.edu/nsnam/ns/.

[14] S. Chachulski et al, "Trading Structure For Randomness in Wireless Opportunistic Routing", Proceedings of ACM Sigcomm, 2007.

[15] Y. C. Hu, A. Perrig, D. B. Johnson, "Ariadne : a secure on-demand routing protocol for ad hoc networks", Wireless Networks, Vol 11, No 1-2 pp 21-28, 2005.

[16] Y.C. Hu, A. Perrig, D. B. Johnson, "Packet leashes: a defense against wormhole attacks in wireless ad hoc networks", Proceedings of IEEE Infocom, 2003.

[17] J. Eriksson, S. Krishnamurthy, M. Faloutsos, "TrueLink: A practical countermeasure to the wormhole attack in wireless networks", Proceedings of ICNP, 2006.

[18] R. Maheshwari et al, "Detecting Wormhole Attacks in Wireless Networks Using Connectivity Information", Proceedings of IEEE Infocom, 2007.

[19] J. Le, J. S. Liu, D. M. Chiu, "DCAR: distributed coding-aware routing in wireless networks", Proceedings of ICDCS, 2008.

[20] M. Chuah, P. Yang, J. Han, "A ferry-based Intrusion Detection Scheme for Sparsely Connected Ad Hoc Networks", First Workshop on Security for Emerging Ubiquitous Computing, Aug, 2007.

[21] J. Dong, R. Curtmola, R. Sethi, C. Nita-Rotaru, "Toward Secure Network Coding in Wireless Networks: Threats and Challenges", Proceedings of Fourth Workshop on Secure Network Protocols, NPSEC, Oct, 2008.

[22] M. Chuah, P. Yang, "Comparison of Two Intrusion Detection Schemes for Sparsely Connected Ad Hoc Networks", Proceedings of IEEE Milcom, Oct, 2006.

[23] M. Chuah, P. Yang, Y. Xi, "How Mobility Model Affects the Design of Network Coding Schemes in DTNs", Lehigh CSE Technical Report, LU-CSE-08-004, Sept, 2008