Flexible and Dynamic Network Coding for Adaptive Data Transmission in DTNs

Milena Radenkovic
Department of Computer Science
University of Nottingham
Jubilee Campus, Wollaton Road
Nottingham, NG8 1BB, UK
mvr@cs.nott.ac.uk

Sameh Zakhary
Department of Computer Science
University of Nottingham
Jubilee Campus, Wollaton Road
Nottingham, NG8 1BB, UK
szz@cs.nott.ac.uk

Abstract - **Existing network coding approaches for Delay-Tolerant Networks (DTNs) do not detect and adapt to congestion in the network. In this paper we describe CafNC (Congestion aware forwarding with Network Coding) that combines adaptive network coding and adaptive forwarding in DTNs. In CafNC each node learns the status of its neighbours, and their ego-networks in order to detect coding opportunities, and codes as long as the recipients can decode. Our flexible design allows CafNC to efficiently support multiple unicast flows, with dynamic traffic demands and dynamic senders and receivers. We evaluate CafNC with two real connectivity traces and a realistic P2P application, introducing congestion by increasing the number of unicast flows in the network. Our results show that CafNC improves the success ratio, delay and packet loss, as the number of flows grows in comparison to no coding and hub-based static coding, while at the same time achieving efficient utilisation of network resources. We also show that static coding misses a number of coding opportunities and increases packet loss rates at times of increased congestion.**

I. INTRODUCTION

Delay Tolerant Networks (DTNs) opportunistically exploit contacts among mobile nodes to allow end to end communication between points that often do not have end to end connectivity. There has been a proliferation of interest in applying network coding to DTNs to improve data transmission efficiency. State of the art routing approaches exploit the power-law behaviour that these networks exhibit (i.e. existence of more centralised nodes that can collectively connect to the rest of the nodes [1]) and use highly central nodes as message relays that employ random linear coding addressed to the same destination [2]. At times of increasing traffic demands, the highly central nodes and parts of the network can get congested. Current approaches do not cope well with congestion because they are not adaptive and result in increased dropped-packet rates and delays. Previous studies have shown that dropping network coded packets can significantly degrade the delivery success rates [3]. In this paper we aim to support congestion aware data transmission with flexible and dynamic deployment of opportunistic per-content network coding in heterogeneous DTNs. Each node uses combined local heuristics on social, resource and content metric to detect and exploit coding opportunities. More specifically, based on its estimate of congestion levels and content distribution, each intermediary node decides on 1) whether it will perform network coding, 2) what the optimal next hop for that content will be, 3) and in what order the messages are sent.

We propose to unify adaptive forwarding and opportunistic network coding into a framework (CafNC) for DTNs that manages to both increase the network coding rate to efficiently utilise resources and to decrease the network coding rate when the resources are too low, in order to avoid packet loss. We achieve this by using a local based implicit network coding utility heuristic that builds on our previous work, which defined contact, interest and resource heuristics in order to adaptively forward and replicate in DTNs [4, 5, 6]. More specifically, in this paper we propose to dynamically combine three types of heuristics: 1) node forwarding predictor, which employs contact analysis in order to develop heuristics to allow optimal directionality and delivery probability of a node; 2) node and ego-network resource driven heuristics that adapt to the nodes' or parts of the network buffer availability, delays or congesting rate; and 3) node interest driven heuristic, which ensures that the content is forwarded towards the nodes with similar interests or who encounter nodes interested in the message topic, provided that they have not already received these messages. The goal of CafNC is to adaptively change between messages being coded and forwarded vs. simply forwarding uncoded messages. CafNC achieves this by using the combined heuristic of relative utility calculations that manages the trade off between the multiple contacts and resource attributes of nodes in real network scenarios.

We show detailed evaluation of CafNC across two CRAWDAD [7] real connectivity traces with different connectivity patterns. We use a publish-subscribe podcasting application for data transfer and we introduce congestion by increasing the number of unicast flows in the network. We present a design overview and extensive evaluation of CafNC at times of increasing congestion, against benchmark, adaptive state of the art DTN protocols and hub-based network coding DTN protocols, across a number of metrics such as packet loss, success rate, network coding cost and delays.

Our results show three main contributions:

1) CafNC achieves higher success ratio for a range of congestion levels than DTN benchmark, adaptive replication algorithms and adaptive single-copy forwarding without coding while managing significantly lower delays, packet loss and network coding cost across two real connectivity traces and

2) we show that adaptive CafNC is able to improve the delivery ratio by 17-43% compared to the static hub based coding solution while having significantly lower cost, packet loss and delays.

3) As CafNC is fully opportunistic and relies on local observations only, it is important to show how often coding opportunities arise in realistic settings and whether they can be detected using only local information. Our results show that CafNC is much better at detecting and utilising the right amount of coding and efficient network utilisation at the right places than the pre-determined statically optimised distributed coding solutions.

This paper is organized as follows. Section II provides a review of the related work on distributed network coding and congestion control in DTNs; Section III describes CafNC framework and the utility heuristics that drive the network coding layer; Section IV discusses our evaluation methodology and shows results; Section V gives conclusions.

II. Related Work

This section discusses a number of newly proposed network coding approaches and methods that allow for intelligent and adaptive forwarding and message replication in DTNs.

[8] models the joint optimisation of congestion control, opportunistic routing and network coding in wireless mesh networks as a network utility maximisation problem. [8] provides formal foundation and useful insights in the fundamental design of real protocols.

[9] proposes an opportunistic approach to network coding, where each node snoops on the medium, learns the status of its neighbours, detects coding opportunities, and codes as long as the recipients can decode. This paper has multiple drawbacks such as that the authors setup static routes to ensure the two senders transmit via the relays and do not exploit knowledge about social and interest networks.

[2] argues that content-based network coding helps in finding the optimal dissemination policy for social structured scenario in DTNs. The paper identifies a number of issues that require further investigation including how to use the distance in the social structure between relays and destinations in coding decisions. Our paper addresses this question that has still not been very well explored by the research community.

[1] proposes a forwarding strategy called HubCode that uses highly central nodes as message relays. [1] assumes that a fraction of top central nodes are collectively connected to the rest of the nodes. While the authors claim that the use of hubs as relays, ensures that most messages are delivered to the destinations. We show that at times of increasing congestion levels, these hubs become overloaded, start dropping network coded packets and achieve significantly lower delivery ratios.

Zhang at al. in [10] investigates the benefits of using Random Linear Coding (RLC) for unicast communications in DTNs under epidemic routing. Under RLC, DTN nodes store and then forward random linear combinations of packets as they encounter other DTN nodes. Through simulation they show that RLC achieves smaller block delivery delay than non-network coded packet forwarding under bandwidth constraint, and the relative benefit increases further when buffer space within DTN nodes is limited.

[11] proposes to improve epidemic routing with ACKs, probabilistic forwarding, location forwarding and consideration of data sizes and introduce two kinds of coding, complete coding and partial coding.

[3] focuses on evaluating the impact of selective data dropping attacks on the delivery performance of a network coding scheme. The results in [3] shows that message dropping in as few as 12% of nodes prevents all traffic traversing the network, leaving direct transfer as the only method of communication between nodes.

An alternative method to network coding is message replication. [12] shows that the number of replicated copies for each message needs to be adaptive. [12] develops a dynamic, local approach to detect and respond to congestion by adjusting the copy limit for new messages. In their work DTN nodes use implicit indicators to detect congestion based on gathered network metrics from their contacts with other nodes. This work has assumed a uniform network with random waypoint mobility. In reality the networks are likely to be non-uniform and the level of congestion may vary between different regions of the network.

In [4, 5, 6] we propose and examine several combined social and resources heuristics in order to detect congested parts of the network and move the traffic away towards less congested parts. The total combined utility function we propose is at the core of our adaptive forwarding protocol that is dynamic and flexible as it operates as a pure social (contact driven) protocol at times of low congestion but is highly resource driven at times of high congestion. We show that our congestion control framework increases performance both when single copy forwarding and replication forwarding, against both benchmark and state of the art forwarding algorithms. In [13] we build an interest-driven P2P content dissemination overlay on the top of our congestion aware forwarding protocol. Both caching and forwarding policies are decided based on the interest, availability, social closeness and numbers of interested nodes. Our results showed that our adaptive overlay manages to maintain high success ratio of answered queries, high availability of intermediary nodes and short download times for a P2P file casting application running on it in the face of increasing number of file publishers and topic popularity. In this work we focus on our frameworks suitability for adaptively controlling network coding, rather than replication.

III. Flexible Network Coding for Congestion Aware Data Transmission In DTNs (CafNC)

This section describes unified adaptive forwarding and adaptive network coding management approach for congestion aware data transmission in DTNs (CafNC).

In our previous work in [19] we have used Erasure Coding (EC) and distributed replication at Intermediaries nodes to defend against malicious dropping attacks in DTNs. Static nodes on the forwarding path between the source and destination were assumed to continuously perform erasure coding to minimize the impact of malicious dropping. However, static network coding techniques cannot support efficient communication in the face of varying connectivity, mobility and application patterns. Contrary to our earlier work, CafNC extends the state of the art network coding in DTNs to be more dynamic and flexible in order to decrease dropping messages, particularly those that are network coded, as message dropping is especially harmful to coded packets

[3]. It is essential that coding is not over restricted, as missing encoding opportunities may potentially cause increased delays and lower success ratios. In order to further decrease delays and to increase success ratios, CafNC aims to encode together messages that share the same topic or are sent to the same receiver.

Rather than choosing that network coding is performed statically on a pre-chosen set of highly central nodes or demanding that nodes perpetually perform network coding [1], we propose that network coding is performed on any node which determines that it has sufficient resources and packets to do the network coding. By allowing the network coding to be performed dynamically, CafNC adaptively prevents network coding in the parts of the network that have low buffer availability, increased node delays and little or no interest in the content, as well as performs network coding in the parts of the network with higher buffer availability, lower node delays, slower congesting rates and with a greater level of interest in the content.

Selecting which node represents the best carrier for the set of messages and deciding whether to network code them are both multiple attribute decision problems where the aim is to select the node that provides the maximum utility for carrying certain messages and only code messages if the next hop is interested in the content and is capable of accepting a coded message without becoming overloaded.

$$Caf\acute{e}Util_D\big(X, C_i(X)\big) = \frac{1}{|H|} \cdot \sum_{h \in H} Util_h(X, C_i(X))$$

(1)

Formula 1 shows the $Caf\acute{e}Util_D$ utility that is the sum of the set of utilities of the following heuristics for a given destination D:

$$H = \{Ret, Rec, CR, FWD_D, WEN_{Ret}, WEN_{Rec}, WEN_{CR}\}$$

H comprises of the following comparable heuristics: Retentiveness (Ret) refers to percentage of a node available storage. Receptiveness (Rec) refers to the in-network delays a node adds when being on the routing path. Congesting rate (*CR)* refers to how fast a node is likely to fill in its buffer space. Ego network of a node refers to all the contacts that node has met in the past. We define ego-network retentiveness, receptiveness and congesting rate ($WEN_{\mathrm{Ret}}, WEN_{\mathrm{Rec}}, WEN_{CR}$) as weighted exponential moving average of nodes' retentiveness, receptiveness and congesting rates respectively. FWD_D refers to SimBetTS [14] social forwarding heuristic. The detailed descriptions of $Ret, Rec, CR, WEN_{\mathrm{Ret}}, WEN_{\mathrm{Rec}}$ and WEN_{CR} are given in [4, 5, 13, 6]. $Caf\acute{e}Util_D$ utility (given in Formula 1) is responsible for capturing the overall improvement a node represents when compared to an encountered node across all measures, when choosing the next hop, such that it best manages tradeoffs across multiple contact and resource attributes of nodes in dynamic and unknown network conditions. Relative utility function for a heuristics h comprises of measurements of relative gain, loss or equality, calculated as pair-wise comparison between the node's own heuristic parameters h(X) and that of an encountered contact h(Ci(X)). This is given in Formula 2.

$$Util_h = \frac{h(C_i(X), X)}{h(X) + h(C_i(X))}$$

(2)

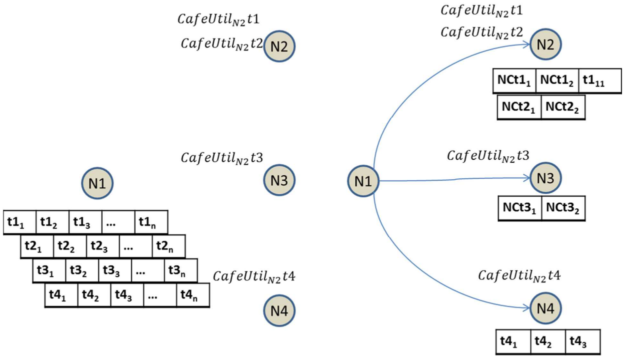
Besides propagating resource and forwarding predictor information, nodes also propagate their interest networks. In order to bridge the gap between interest and resource layers, the following three steps need to be performed when the two nodes meet: first, recipients interested in the messages are determined (as in [13]), we assume that they can be either the nodes that directly interested in the content or have strong ties with the nodes that are interested in that content; second, messages $m_{i..j}$ are determined that can be network coded into a message *M*; Finally, the candidate nodes with the highest probability of delivery for coded message *M* or non-coded messages $m_{i..j}$, while also avoiding congested regions is selected and the messages are transmitted.

By using Café's congestion aware forwarding heuristic we are able to code more packets when the level of congestion is low and therefore the risk of packet loss is low, and reduce the coding intensity when the risk of packet loss is high. By using ego-network resource utilities, CafNC is able to provide a wider view of the network resources.

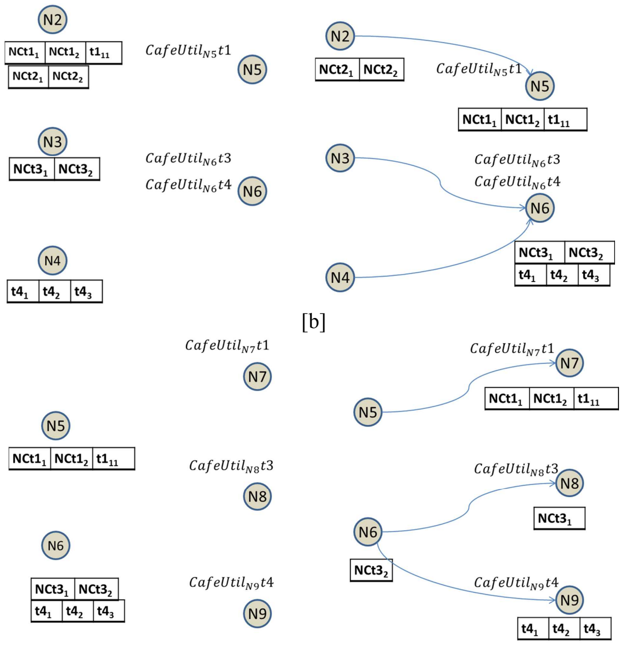
$$NCRate = \lfloor Caf\acute{e}Util_D(X, C_i(X)) + 1 - CafNCThreshold \rfloor$$

(3)

Formula 3 shows how we dynamically calculate whether messages should be coded when forwarding or not. We evaluate$Caf\acute{e}Util_D$ against *CafNCThreshold that is* a configurable parameter between 0 and 1. When this parameter is high, CafNC reduces the amount of coding a node carries out, and when it is low it relaxes the coding criteria. This results in CafNC detecting and exploiting more coding opportunities, on a wider range of nodes, when congestion is low in comparison with static coding techniques and adaptively codes less at times of extreme congestion, so that packet loss rates are significantly reduced. We allow the sender to stop sending until it finds the right node that it can redirect the traffic to without incurring additional packet loss.



[a]



[b]

[c]

Figure 1. Multi-Party Application Examples

Consider an example of a multi-party application, given in Figure 1, in which there are a number of publishers and subscribers distributed around a network, and where each subscriber wishes to receive the audio streams from all of the publishers (senders can send either via unicast or multicast). Congestion aware network coding can be performed by a number of highly centralised nodes (access points or road side units) that already exist in the network or by any nodes in the network that opportunistically discover that they have the content and resources to code certain packets and send to the next hop. Figure 1 shows how four flows that traverse different paths can be encoded.

In Figure 1 (a), Node *N1* has 4 topics (*t1*, *t2*, *t3*, *t4*) that each comprise of *n* messages. Node *N1* meets nodes *N2*, *N3*, *N4*. Node *N2* is best fit for topic *t1* that gets partially network coded, and topic *t2* that gets fully network coded. Node *N3* is best fit for topic *t3* that gets fully network coded. Node *N4* is best fit for topic *t4*, messages get forwarded as they are – not network coded.

In Figure 1 (b) Node *N2* with topics *t1*, *t2*, Node *N3* with topic *t3*, Node *N4* with topic *t4* meet Nodes *N5* and Node *N6*. Node *N5* is best fit for topic *t1* that gets partially network coded. Node *N6* is best fit for topic *t3* that gets fully network coded and topic *t4* where messages get forwarded as they are not network coded. No nodes are fit to receive topic *t2* from Node *N2*.

In Figure 1 (c), Node *N5* with topic *t1* and Node *N6* with network coded topic *nct3* and topic *t4* meet Nodes *N7*, *N8* and *N9*. Node *N7* is best fit for the partially network coded topic *pnct1*. Node *N8* is best fit for the fully network coded topic *nct3*. Node *N9* is best fit for topic *t4*.

```
Algorithm 1
For all Topic ∈ Buffer do
  sortedContacts ← currentContacts sorted by CafeUtil$_D$ * length(Topic)
  Forr all Contact C$_i$ ∈ sortedContacts do
    If NCRate(X, C$_i$) == 1 and length(Topic) ≥ NCLimit
    then
      send NCCode(Topic(C$_i$))
    Else
      Send Topic(C$_i$)
    End If
  End For
End For
```

Figure 2. CafNC Message Transfer Algorithm

Figure 2 gives an overview of the pseudo code for our CafNC that works as follows: For each topic in the node buffer, the node scans for neighbouring nodes interests and calculates their respective relative $CaféUtil_D$. Each neighbouring nodes $CaféUtil_D$ is weighted by the number of spaces for the respective topic that node has. The sending node then sorts the weighted neighbouring nodes utilities so that the node with the largest weight appears first in the list. The list is then traversed and for each member the number of packets for the respective topic is compared to the predefined network coding limit *NCLimit*. If the number of packets for the topic exceeds the threshold, then the node network codes the messages in the topic in sets of *NCLimit* and sends its coded messages to the corresponding neighbouring node, if there are enough messages in the topic set to code CafNC forwards the messages without coding them. The value of *NCLimit* is configurable. In this instance we have chosen to set it to a value of 5.

$$X_{n_i}(t)=\sum_{i=1}^{i=M} g_i(t)P_i+\sum_{j=1}^{j=N} h_j(t)Q_i \qquad (4)$$

Formula 4, motivated by [2], shows that messages are encoded at time t at node n_i by combining the packets received $(P_1,...,P_M)$ and the messages within its buffer $(Q_1,...,Q_N)$ with the coefficients $g_i(t)$ and $h_j(t)$ respectively. The sending node generates a random vector and employs it to do a linear combination of the packets cached that are targeted to the same destination. The destination nodes will collect packets which are linearly independent and as soon as the number of these packets reaches a certain number, the destination node will decode them and deliver them to the upper layer.

Figure 3 illustrates the functional architecture of a CafNC, which comprises of the following elements:

- Buffer/Input Queue, which receives and queues messages and network coded messages from neighbouring nodes.
- Topic Selector, that determines which of the stored messages should be forwarded as is and which should

be mixed before forwarding in order to meet a bandwidth target specified by the Next hop selector.

- Network Coder, which takes designated messages from the packet selector and produces network coded messages.
- Transmitter, which sends coded and uncoded messages on to the next hop (intermediaries or destination)
- Congestion monitor, which estimates in-network delays, buffer levels and congestion rates in nodes and their ego-networks for the previously transmitted messages, based on the contact and resource information exchange with the potential next hop neighbouring nodes.
- Nest hop selector, which weights the current contact set and ranks them to determine the optimal next hop and the target sending bandwidth, in order to be adaptive to congestion.

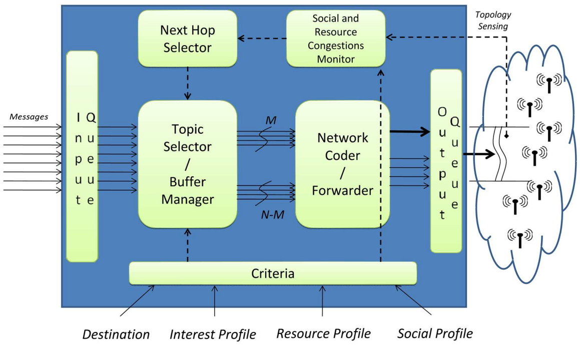
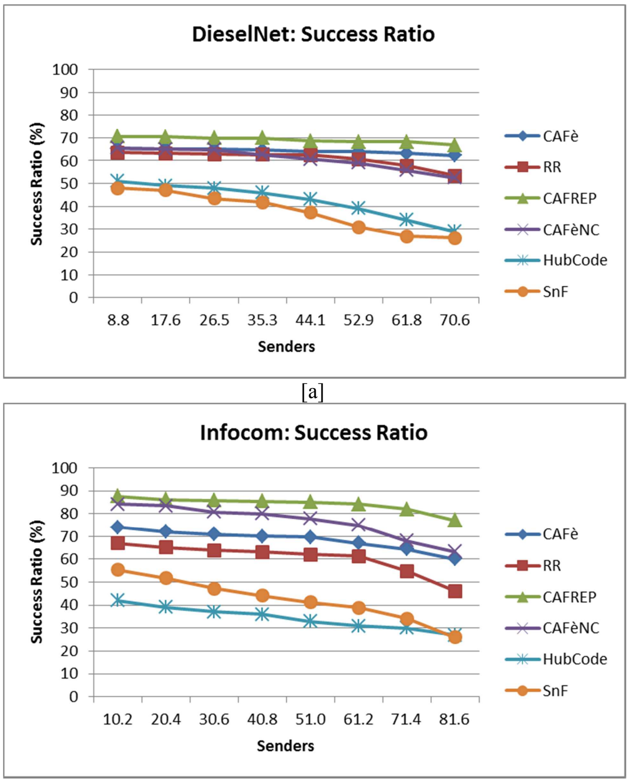


Figure 3. The functional architecture of a CafNC

IV. Evaluation

We perform an extensive evaluation of CafNC in comparison to five protocols: an adaptive single copy forwarding algorithm (Café [6]), adaptive multiple-copy forwarding algorithms (CafRep [4], Retiring Replicants [12]), a non-adaptive multi-copy forwarding algorithm (Spray and Focus [15]) and a static network coding algorithm (HubCode [1]), over multiple criteria using two vastly different connectivity datasets, Infocom 2006 [16] and DieselNet [17], from the CRAWDAD wireless data archive. The Infocom 2006 dataset [16] consists of a 4-day long trace that is based on a human mobility experiment conducted at Infocom 2006. A total of 78 volunteers joined the experiment and each was given an iMote device capable of connecting to other Bluetooth-capable devices. In addition 20 static long-range iMote devices were placed at various locations of the conference venue; three of these were semi-static as they were placed in the building lifts. The DieselNet dataset [17] consists of 20 days of traces of 40 University of Massachusetts transit buses covering approximately 150 square miles. This trace contains connection events between busses as well as between buses and Access Points. DieselNet buses were subject to the schedule of the University of Massachusetts campus. This trace exhibits long periods of disconnections, short periods of connectivity and islands of connectivity that are rarely populated by more than two nodes.

Our content is organised as in previous file-casting work [18, 13]: Each chunk has a unique ID and the topic has the total number of chunks. Nodes randomly choose to be interested in a certain topic. Each node has a queue size of 1000 units. Podcasting nodes send at the rate of 5 chunks a second. We have run eight experiments with increasing number of publishers in steps of approximately 10% of the total experiment participants. All simulations are repeated five times with different random subscribers and publishers. In this paper we report on experiments with increasing number of publishers due to the lack of space but note that the results for increasing number of subscribers are similar to those presented here.



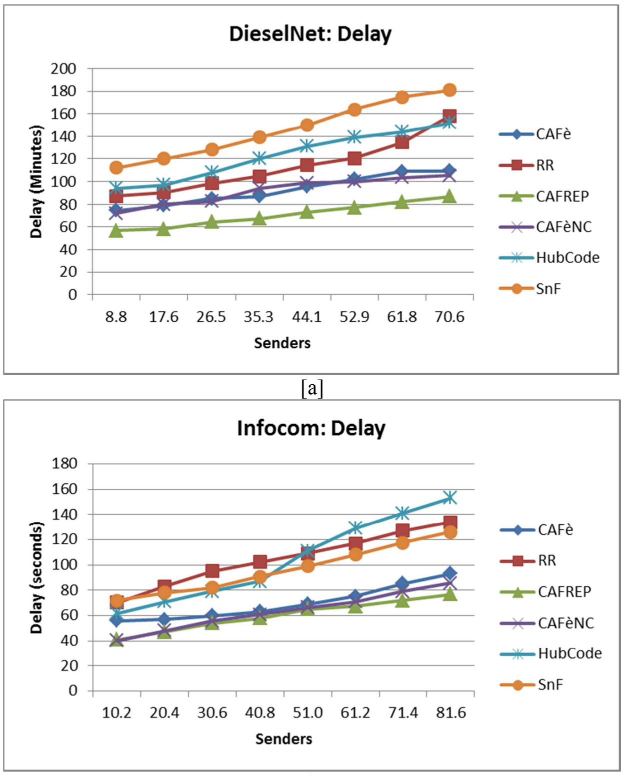
[a]

[b]

Figure 4. Success Ratio

Figures 4a and 4b depict in percentages the protocols' success ratios on the vertical axis in the face of increasing congestion levels. It shows that, for both scenarios, CafNC can dramatically improve the success ratio of congested DTNs, in comparison to HubCode.

We observe that in Figure 4a (DieselNet) Spray & Focus and HubCode have the lowest success ratios of 37.7% and 42.3% correspondingly. CafNC (60.7%) and RR (60.8%) perform similarly and show 67%-98% improvement in comparison to SF and HubCode. Café with success ratio of 64.2% performs similarly to CafNC when congestion is low, but better maintains its success ratio as congestion increases. CafRep has the highest success ratio, with only between 7% and 10% difference to CafNC in low and high levels of congestion respectively. Figure 4 (b) (Infocom) shows that using congestion aware forwarding (Café) with network coding and replication has positive impact on the success ratio of messages in social scenarios, with CafNC at 76.5% performing better than Café (68.5%), RR (60.4%), SF (42.3%) and HubCode (34.3%) over the Infocom 2006 dataset. The figure demonstrates that in social settings HubCode has the weakest performance. CafRep outperform CafNC by 5-15%. CafRep outperforms CafNC because of two reasons: first, CafRep has lower packet loss rates and second, CafRep has more accurate replication rate calculation compared to network coding rate as it has no fundamental limit such as NCLimit.



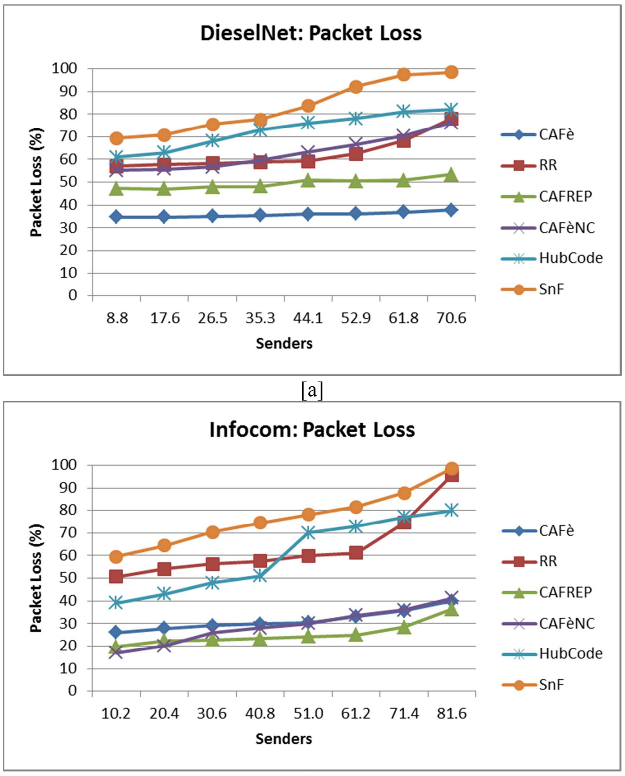
[a]

[b]

Figure 5. Delay

Figures 5a and 5b show protocols' delivery delays in the face of increased congestion levels for two realistic traces. As congestion increases, Café chooses more hops as it goes in a round-about way to the destination, avoiding congested nodes and regions. Figure 5a shows that despite the increase in path length, CafNC experiences similar delays (87.9 min) as no coding (Café 90.6min) but substantially lower delays than HubCode (123.1min). Only CafRep outperforms CafNC due to finer replication rate. CafNC outperforms Sf (by up to 100 minutes) and RR (by 40 minutes). Figure 5b (Infocom) shows that there is little difference in delay between CafNC (63.3 sec) and CafRep (60.1sec), as they outperform all other algorithms: Café (69.7sec), SnF (96.7sec), HubCode (104.1sec), RR (104.6sec).

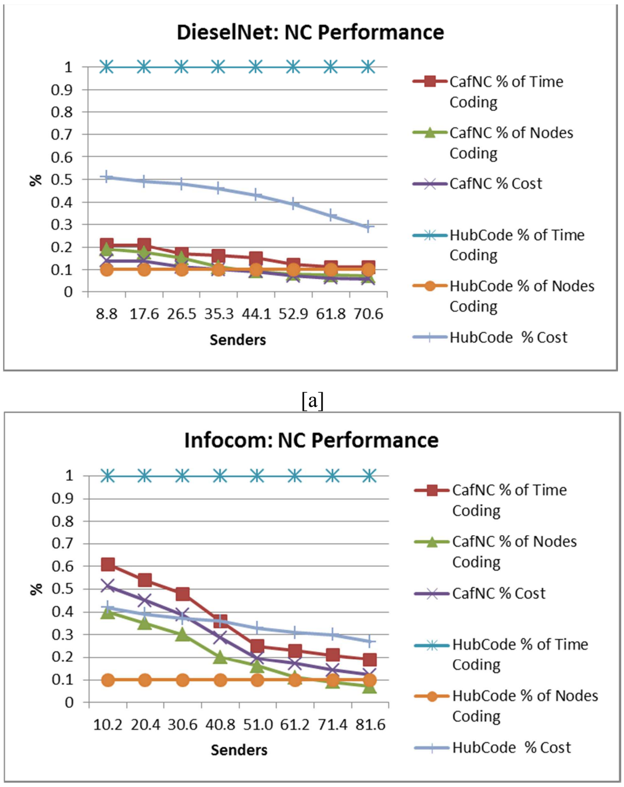
Figures 6a and 6b show the percentage of packet loss that protocols experience in the face of increased congestion levels. Figure 6a (DieselNet) shows that Café has the lowest packet loss of 35% followed by CafRep (49%), RR (62.4%), CafNC (62.9%), HubCode (78%), SnF (83%). Figure 6b shows that, for low levels of traffic and within the social connectivity setting, packet loss is lower for CafNC (18%) than for any other method of forwarding. As the level of congestion increases, packet loss rates for CafNC become more in line with the single copy uncoded approach CAFÉ) and up to 15% higher than for CAFREP. The lower loss rates for CafRep are due to finer granularity calculation of replication rate (versus network coding rate) that does not have preconfigured parameters and is more flexible in adjusting its transmission rate than CafNC.



[a]

[b]

Figure 6. Packet Loss



[a]

[b]

Figure 7. Network Coding Performance

Figures 7a and 7b show the percentage of time and percentage nodes that perform network coding for adaptive (CafNC) and hub-based static (HubCode) network coding in the face of increased congestion levels. For DieselNet, Figure 7a shows that CafNC uses 11% of nodes to encode traffic 15% of the time at a total cost of 9.5% where HubCode uses 10% of the node to encode traffic all of the time at a total cost of 42.3%, which is 400% worse off than CafNC. For Infocom, Figure 7b shows that CafNC uses 21% of the nodes to encode traffic 35% of the time at a total cost of 28% where Hubcode uses 10% of the nodes to encode traffic all of the time at a total cost of 34.3%. We observe that HubCode misses many coding opportunities when congestion is low and codes at a similar rate when the risk of packet loss is elevated. HubCode selects 10% of nodes to code all the time, but as a result HubCode misses the other opportunities to save on transfer bandwidth. Also, at times when the elected hubs are congested, HubCode has significantly higher packet loss, which makes coding at these nodes a costly strategy.

V. Conclusion

We proposed CafNC that uses local resource, social and interest relative utilities for adaptive congestion aware network coding. CafNC significantly increases success ratio while decreasing latency and packet loss rates at times of increasing congestion levels compared to hub based coding in DTNs, adaptive replication (RR), DTN benchmark routing protocol (SF) and congestion aware forwarding (CAFÉ). CafRep outperforms CafNC because of the replication rate calculation is finer grained than network coding rate calculation which has to take into account the minimal number of messages coded together (NCLimit). Similarly, CafNC has higher packet loss rates than CafRep because losing network coded packets with single copy of the packets is fundamentally more detrimental than losing packets where there are multiple copies of the packets.. Our results show that as congestion increases, without an adaptive amount of network coding, the performance of network coding in DTNs deteriorates quickly because of the high level of contention in the network. In contrast, with the right amount of coding, the number of transmissions reduces for the same amount of data, resulting in lower congestion and consequently better performance. Our future work will look into extending CafNC in order to allow finer grained network coding rate calculation and scheduling of packets based on more profound social and content analysis.

Acknowledgment

This work was supported by the Engineering and Physical Sciences Research Council UK (EPSRC) Grant number EP/D062659/1

References

[1] S. Ahmed and S. Kanhere, "Hubcode: message forwarding using hub-based network coding in delay tolerant networks," in *Proceedings of the 12th ACM international conference on Modeling, analysis and simulation of wireless and mobile systems* ACM, 2009, pp. 288–296.

[2] G. Karbaschi, A. Viana, S. Martin, and K. Al Agha, "On using network coding in multi hop wireless networks," in *Personal, Indoor and Mobile Radio Communications, 2009 IEEE 20th International Symposium on*. IEEE, pp. 420–424.

[3] M. Chuah and P. Yang, "Impact of selective dropping attacks on network coding performance in dtns and a potential mitigation scheme," in *Computer Communications and Networks, 2009. ICCCN 2009. Proceedings of 18th International Conference on*. IEEE, 2009, pp. 1–6.

[4] M. Radenkovic and A. Grundy, "Framework for utility driven congestion control in delay tolerant opportunistic networks," in *the 7th International Wireless Communications and Mobile Computing Conference (IWCMC 2011)*, Istanbul, Turkey, Jul. 2011.

[5] M. Radenkovic and A. Grundy, "Congestion aware forwarding in delay tolerant and social opportunistic networks," in *Eighth Annual Conference on Wireless On demand Network Systems and Services (WONS)*, 2011.

[6] A. Grundy and M. Radenkovic, "Promoting congestion control in opportunistic networks," in *IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob'2010)*, Niagara Falls, Canada, Canada, 10 2010.

[7] D. Kotz and T. Henderson, "Crawdad: A community resource for archiving wireless data at Dartmouth," *IEEE Pervasive Computing*, pp. 12–14, 2005.

[8] H. Seferoglu and A. Markopoulou, "Network coding-aware queue management for unicast flows over coded wireless networks," in *Network Coding (NetCod), 2010 IEEE.*

[9] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, and J. Crowcroft, "Xors in the air: practical wireless network coding," in *Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications*. ACM, 2006, pp. 243–254.

[10] X. Zhang, G. Neglia, J. Kurose, and D. Towsley, "On the benefits of random linear coding for unicast applications in disruption tolerant networks," in *Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks, 2006 4th International Symposium* IEEE, 2006.

[11] Q. Zhang, Z. Jin, Z. Zhang, and Y. Shu, "Network coding for applications in the delay tolerant network (dtn)." in *MSN'09*, 2009, pp. 376–380.

[12] N. Thompson, S. Nelson, M. Bakht, T. Abdelzaher, and R. Kravets, "Retiring replicants: Congestion control for intermittently-connected networks," in *Proceedings of Infocom*, 2010.

[13] M. Radenkovic and A. Grundy, "Poster: Congestion Aware Data Dissemination in Social Opportunistic Networks," *SIGMOBILE Mobile Computing and Communications Review (MC2R)*, vol. 14, no. 3, pp. 31–33, 2010.

[14] E. Daly and M. Haahr, "Social network analysis for information flow in disconnected Delay-Tolerant MANETs," *IEEE Transactions on Mobile Computing*, pp. 606–621, 2008.

[15] T. Spyropoulos, K. Psounis, and C. Raghavendra, "Spray and focus: Efficient mobility-assisted routing for heterogeneous and correlated mobility," in *Pervasive Computing and Communications Workshops, 2007. PerCom Workshops' 07. Fifth Annual IEEE International Conference on*. IEEE, 2007, pp. 79–85.

[16] J. Scott, R. Gass, J. Crowcroft, P. Hui, C. Diot, and A. Chaintreau, "CRAWDAD data set cambridge/haggle (v. 2009-05-29)," Downloaded from http://crawdad.cs.dartmouth.edu/cambridge/haggle.

[17] N. Banerjee, M. Corner, D. Towsley, and B. Levine, "CRAWDAD data set umass/diesel/transfer/fall2007 (v. 2008-09-14)," Downloaded from http://crawdad.cs.dartmouth.edu/umass/diesel/transfer/fall2007.

[18] M. May, V. Lenders, G. Karlsson, and C. Wacha, "Wireless opportunistic podcasting: implementation and design tradeoffs," in *Proceedings of the second ACM workshop on Challenged networks*, ACM, 2007.

[19] Zakhary, S.R. and M. Radenkovic. Erasure Coding with Replication to Defend Against Malicious Attacks in DTNs. in WiMob 2011. IEEE.