Quality of Service Enforcing Centrally Optimized Routing for Delay Tolerant Networks

Abhishek Thakur, Chittaranjan Hota
Computer Science and Information Systems
BITS Pilani, Hyderabad Campus
Hyderabad, India
{abhishek, hota}@hyderabad.bits-pilani.ac.in

***Abstract* — Central node(s) that collect global view on delay tolerant network performance can simplify management and ensure good quality of service. Such global view can improve DTN routing without increasing the routing complexity on individual nodes. Existing research using central nodes have explored both routing control and routing relay via central node. This paper proposes central node assisted routing where message forwarding and relay continues in pure distributed manner. Global view computed by central node augments the routing decision on individual nodes.**

Additionally this paper evaluates the extra overhead and congestion effects when all application messages are relayed through central node (for policy enforcement, management etc.).

Keywords— Delay Tolerant Networks, Quality of Service, Fairness, Adaptaton, User Generated Content

I. INTRODUCTION

Delay and disruption tolerant networks (DTN) [1] come into play when end-to-end communication using mesh or ad-hoc networks are not feasible. Node mobility is the primary enabler for DTN as it emphasizes on store-carry-forward behavior for the bundles (payload or messages for DTN).

Authors in their prior work, [2], have attempted to deploy DTN based platform for rural communication with focus on applications for intra-community communications. This is a scenario where DTN based terrestrial communication service is being provided on a long term basis for the village community. In such deployments people with low mobility or nodes in the periphery of the network are treated unfairly. It is observed that they have lower delivery ratios and/or significantly higher delays.

Revenue generation is essential to make DTN rural deployment sustainable. Any revenue model mandates features like - accounting, billing, security, compliance to local laws and quality of service. In this context, challenges in pure distributed DTNs are a) lack of metering and accounting of resource usage, b) tracking and enforcement of quality of service for individual nodes, c) enforcement of policies to comply with local laws, and d) ability to do security scans (like IDS/IPS in commercial TCP/IP based networks).

To mitigate the challenges of pure distributed DTN, the authors, in their prior work have explored deploying a central device to simplify user management, billing, policy enforcement etc. The owner / operator of the central device will be able to transparently bill the end users for the services they have received. Central device, referred to as CoHub (centrally optimizing hub) in this paper, creates a global view of the community DTN nodes. Such a global view on CoHub can drive the routing protocols on individual nodes to target specific quality of service for end nodes.

One of the applications for this rural platform is sharing highlights from farm monitoring videos. Rather than sending complete video, only masked highlights are sent (e.g. video snippet at goo.gl/hc9MuY or goo.gl/s5XA8l). Since video generates large content (megabytes) on periodic basis (30 seconds to few minutes), hence it is important to know the DTN quality of service, so that the highlight creation and compression can be adapted to the network conditions. Information shared by CoHub can drive such decisions.

This work proposes a new routing extension "Quality of Service Enforcing Centrally Optimized Routing", referred to as QoseCo in rest of the paper. In the present work, the authors use Spray-and-Wait [3] as the base protocol, on top of which QoseCo extension is demonstrated. Performance of QoseCo + Spray-and-Wait is compared with vanilla Spray-and-Wait (SNW). Message forwarding is done by all nodes based on information shared by the CoHub. In absence of CoHub, QoseCo will degrade to vanilla SNW. Other protocols like PRoPHET can also be adapted with QoseCo.

An extension of QoseCo is explored where all application messages are forced to be relayed via the CoHub. This is referred to as QoseCo with Central-Relay [QoseCo-CR]. Though relaying via central nodes is expected to cause congestion and make CoHub the single point of failure, it can be used for security scanning and policy enforcements. Usage of multiple CoHub (horizontal scaling) or vertical scaling are some mitigation aspects to improve QoseCo-CR, but the same is not included in the scope of this paper. This paper only attempts to measure the impact on delivery ratios for QoseCo-CR for a single deployment of CoHub.

Structure of rest of the document is as follows: Section II provides a brief overview of related DTN routing protocols and quality of service related work. Section III provides overview of QoseCo and QoseCo-CR along with algorithmic details. Section IV has simulation setup, results and analysis. Section V includes conclusion and a broad overview of future work targeted by the team of authors.

II. Related Work

Massri and others in [4] provide a good overview of DTN routing. They have identified three major components in DTN routing– 1) queue management, 2) forwarding decision and 3) replication. Cao and others [5] provide a more recent survey on various DTN routing approaches. In context of [5], rural deployment can be seen as suburban networks or pocket switched networks. Routing decisions in such cases can use infrastructure based approach with auxiliary nodes like mule or throw box or use participant nodes for store-carry-forward. In this paper the authors use a hybrid approach wherein "infrastructure" is used to improve routing decisions, but the actual routing is done using participant nodes in a distributed manner.

Some of the newer protocols explore human / social aspects of the person carrying the device to drive the routing decisions. This increases the computation complexity on the nodes as each node needs to have visibility beyond its own contacts to calculate social characteristics like centrality and between-ness. For DTN networks with power-law, [6] explores DTN routing via hub nodes, where the (multiple) hub nodes have higher connectivity and offload computation load from other nodes. Another set of protocols use an oracle or deterministic approach to compute the future contact probabilities and drive the routing decisions accordingly. But when these protocols are applied for pocket switched networks, especially for rural people who need not keep a strict schedule, they perform poorly.

When adding an extension to an existing routing protocol, the base protocol should be as simple as possible. Naïve Spray-and-Wait [3] was the obvious choice as it allows for good control on overheads and is very simple in terms of computation and storage overheads on all nodes. Origin node in SNW computes the maximum number of replicas (L) that will be created. Thereafter some replicas are given to nodes on contacts. When operating in binary mode, half the replica count will be shared with other node on contact. Eventually at max L nodes will have a copy of the message. If any of these nodes come in contact with the destination, the message is delivered.

To target fairness, QoseCo allows higher L to be used for messages with poorly served nodes as origin or destination. During forwarding QoseCo allows higher share of replicas to be given to well-connected nodes. Other than SNW, PRoPHET was also considered. Tolerance limits can be relaxed for poorly served nodes if PRoPHET is used. For brevity, it is not discussed further in this paper.

A. Quality of Service:

Hay and Giaccone [7] have explored QoS for DTN routing when contacts between nodes is known in advance or can be predicted in advance (e.g. Bus routes etc.). Such knowledge is used to model the DTN network as time-independent graph for optimizing QoS. Sandulescu [8] uses special messages under nominal load from the source node to destination node and collects the feedback over time. Thereafter source adapts the transmission to observed network behavior. QoseCo targets to do away with such needs by periodically updating about poorly served nodes to CoHub which shares it to the whole network.

B. Fairness in DTN:

Pujol et. al. [9] attempt to measure fairness in the share of load on individual nodes. Fan [10] et. al. measure the fairness in delivery ratios for increasing number of contacts. During relay of messages their work attempts to locally compute the utility value for each message, with delivery farness of messages as a goal. Since each node tries to compute fairness without a global perspective, the impact is limited.

Similar to [10], this paper attempts fairness in delivery ratios. To the best of knowledge of the authors, control of delivery fairness w.r.t. end nodes has not been dealt with in any prior work. QoseCo uses historical contact and delivery information to adjust the priority of nodes. Since complete graph need not be built on each node, QoseCo scales quite well, though it has a time lag in sharing the global view.

III. QoseCo Routing Extensions

QoseCo: All nodes periodically share their state information with CoHub. Based on the information received, the CoHub generates a consolidated view and updates all nodes (broadcast) with control information. This is a communication overhead (below 2%), but it helps move the computation to CoHub. CoHub periodically distributes the global context to all nodes. In absence of information from CoHub, the node degrades itself to naïve routing protocol (SNW in this case).

QoseCo-CR: If complete control of the deployment (e.g. lawful interception, virus scan etc.) is required, all traffic is relayed through the CoHub.

Major decision points are captured below.

A. New message from origin node

Replication count (L) for the message is set by the origin node, based on the last routing control message received from the CoHub. For poorly served nodes, higher counts are allowed.

Algorithm 1: New Message on originating node (simplified)

```
function newMessage(Message M, Host origin, Host dest)
  L = Default_Max_Replica
  if (dest or origin in poorMap)
    lowQual = MIN(getQuality(dest), getQuality(origin)
    L = L * (1+AverageQuality / lowQual)
  if(routing == QoseCo-CR)
    m.setMetadata("actualDest", dest)
    m.destination = CoHub
  m.setMaxReplica(L)
  m.setDestination(destination)
end
```

For QoseCo-CR, destination is set as CoHub and actual destination is added as metadata to the message. Once CoHub gets the message, it does the required processing and relays it to the actual destination.

B. Feedback to the CoHub

All nodes periodically send status updates to CoHub (management payload). The update message includes a) information about the nodes contacts (contact count and total duration) with other nodes; b) count of messages generated and delivered from/to end applications on the node and c) list of messages received before TTL expiry.

C. Consolidation of data (Global-View) at CoHub

When a CoHub receives status update messages from the nodes, it adjusts the metrics for the nodes. Periodically the CoHub multicasts routing updates to all the nodes. This multicast status includes a) list of poorly served nodes and their priority, b) list of nodes that are good carriers with their priority, and c) list of recently received messages by destination nodes. Poorly served nodes are identified as nodes who have reported very few contacts and/or whose packets have not been delivered (either as origin or destination). Good carriers are nodes that have had large number of contacts with different nodes. Exponential weighted moving average is used to reduce impact of really old contacts.

When nodes get multicast routing updates from CoHub, they delete all messages in their local cache that are successfully delivered to the destination nodes. They cache the details of good carriers and poorly served nodes for subsequent routing decisions and new message creation.

D. Routing decesion: Message forwarding between nodes

During contacts, nodes exchange messages with each other as per following algorithm.

Algorithm 2: Forward Message Between Nodes (simplified)

```
function startTxfr(Msg[] pendingMsgs, Host[] connList)
  if (CoHub in connList &&
       CoHub in getDestinations(pendingMsgs))
     candidateMsgs = filterOnDest(pendingMsgs, CoHub)
     sortOnOriginNodeQuality(candidateMsgs)
     transferAndDelete(candidateMsgs[0])
  else
     trgtMsgs = filterOnDest(pendingMsgs, connectedList)
     if(trgtMsgs!= EMPTY)
       sortOnOriginOrDestQuality(trgtMsgs)
       transferAndDelete(trgtMsgs[0])
     else
       txrList = sort_On_EndNodes_CopyCount_Relay(
                           pendingMsgs, connList)
       relayAndAdjustL(txrList[0].msg, txrList[0].remote)
end
```

The message transfers are attempted as follows: A) Deliver messages with CoHub as destination, if one of the connected nodes is a CoHub. B) Else deliver messages with destination as connected node. C) Else relay other messages and adjust copy counts.

If there is a tie w.r.t. message or connected host, the tie is broken as follows: a) messages moving towards CoHub, b) messages with nodes as source or destination of poor quality, c) current replica values of messages, and d) relay to host which are good carriers. *relayAndAdjustL* gives higher number of copies to nodes that are in "Good Carriers" list.

E. Message deletion when buffers are full

Expired messages are deleted first. If required, further messages are deleted based on priority of end nodes.

IV. Simulation and Results

A. Implementation details

QoseCo routing and forwarding, as discussed in section III above was implemented on ONE [11]. Reporting functions were added to appropriately capture the details of management packets and include measure of fairness metric. Jain's throughput fairness index [12] is used. These metrics are captured in table 1.

TABLE I. Delivery Ratio, Overhead and Fairness

Parameter	Value
T_Delivered	Total messages successfully received at destination
T_Started	Total messages sent
T_Relayed	Count of successful forwards of messages between all nodes
N_Success	Number of messages successfully delivered for a specific node (either as source or destination)
N_Attempt	Number of messages for a node (either as source or destination)
Delivery Ratio	T_Delivered / T_Started
Overhead	(T_Relayed – T_Delivered) / T_Delivered
N_Xfactor	(N_Success/N_Attempt) / DeliveryRatio
Fairness	$(\sum$ N_Xfactor$)^2$ / $\sum$ (N_Xfactor^2) for nodes that participated in attempts

Two mobility models from [11] are used - (a) Working Day Movement model – WDM; and (b) Helsinki City Scenario – HCS. WDM simulates users in home, office and shopping malls etc. in a somewhat realistic scenario based on large area of Helsinki city. HCS simulates people moving randomly (walking or in car) across a smaller area within Helsinki. For each user HCS chooses the next destination randomly based on points of interests in the map. For the chosen destination, shortest path is computed based on maps; while the speed and period of stay are randomized based on configuration settings. Readers are requested to refer Keränen's work [11] for further details about these scenarios.

B. Simulation Configuration

To target rural scenarios number of nodes were significantly reduced and simulations were run for longer intervals. Longer duration was chosen, as compared to [11] because the rural area of developing countries will not have internet connectivity at home / office. For HCS, 37 pedestrians, 3 people with cars or two wheelers and 21 static nodes were used along with three trams. For WDM, numbers from [11] were scaled down by factor of sixteen to get 64 end user nodes. For both the scenarios static CoHub node was added. CoHub location was manually chosen near road junctions to ensure good contact opportunities.

TABLE II. DEFAULT SETTINGS FOR SIMULATION RUNS

Parameter	Value	Parameter	Value
Duration	4 days	NumCopies	64
Event Gap	8 minutes (randomized)	EWMA Alpha (QoseCo)	0.002
Event Message Size	750 KB (randomized)	NumPoorService (QoseCo)	16 nodes
Message TTL	8 hours	NumGoodCarriers (QoseCo)	8 nodes

QoseCo update messages are pretty small (typically 4-16 KB for CoHub routing updates and 1-4 KB for node status updates). They have not been included in further analysis since they are smaller by order of hundred times. For a sample run, 200 megabytes of traffic relay was caused because of QoseCo management load while application payload generated 13 gigabyte of traffic.

C. *Simulation Runs*

From the simulation scenarios that were executed, two reports are presented here. Results for Delivery Ratio, Fairness and Overhead are presented for simulation period. For Inter-event Gap (Congestion) the results are discussed only for fairness and delivery ratio because of space limitations. All metrics are average values for eight randomized runs.

1) Simulation Duration- low load

Execution was done for simulation periods of 0.5 days, 1 day, 2 days, 4 days, 8 days, and 16 days. The results are plotted in Figure 1 for (a) Delivery ratio, (b) Fairness and (c) overheads. As simulations run for longer interval, delivery ratios improve for all scenarios. This is because higher percentages of messages are received within the simulation time. QoseCo follows similar trajectory as SNW with a slight degradation. The degradation is observed because management messages compete with application messages during contact opportunities. QoseCo-CR has lower delivery ratios. This is because the routing via CoHub requires more hops and increases congestion in the vicinity of the CoHub.

As duration increases, fairness improves since the messages are generated with random nodes as source and destination. Both QoseCo and QoseCo-CR show significant improvements till 2-4 days, with the slope tapering off for 8 days and 16 days.

For runs of less than 2 days, lots of messages are not yet delivered; hence overheads are observed to be higher. Overheads are higher for QoseCo-CR as it first sends messages to CoHub and then CoHub relays to destination. Moreover HCS has lot more contacts; hence it creates many more copies. Overhead results for QoseCo were marginally higher as it allowed larger number of replicas for poorly served nodes. All through QoseCo had fairness which was pretty close to SNW, despite the delivery ratio being poorer for QoseCo on account of added management messages.

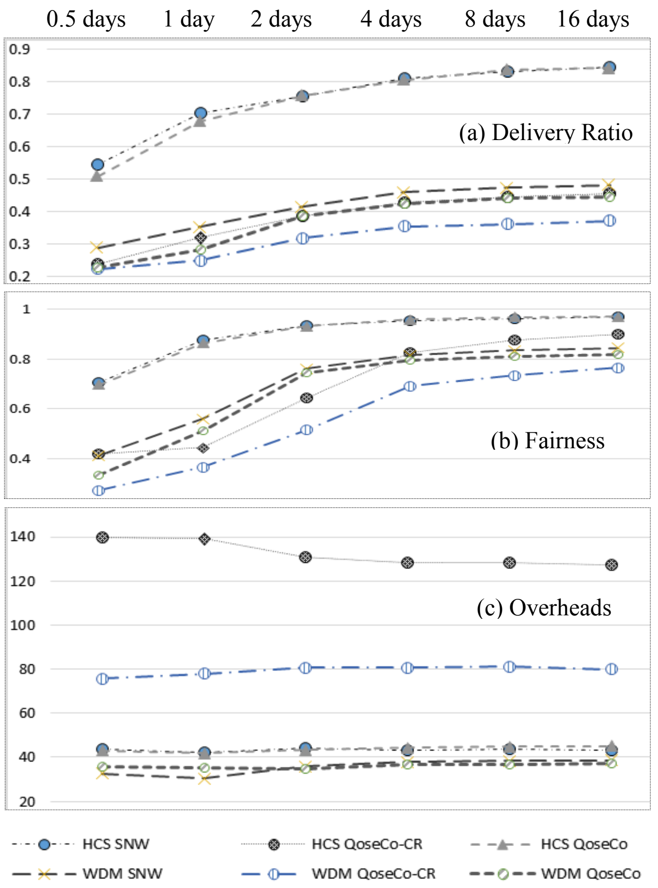

Figure 1 Simulation Duration Results

2) Inter-event Gap- Congestion

Events were randomly generated with average gap of 30 seconds, 1 minute, 2 minutes, 4 minutes, 8 minutes and 16 minutes. This scenario allows for congestion to happen in the cases where messages are generated at two minutes and below. Figure 2 plots the results for the fairness and delivery ratio for four day run.

When events are generated twice every minute, almost all nodes participate in the communication and most of them observe high losses, though the nodes on periphery will be worse off for WDM. For HCS, QoseCo does well to improve fairness in this scenario as it has values around 94% while SNW is around 90%. Similar trend is also observed for WDM (70% vs. 74% for 30 seconds). At lower frequencies since less number of nodes participate in communication, slight degradation is observed in fairness for all cases.

QoseCo-CR observes significantly higher packet loss when message frequency is high (low delivery ratio). This is because the messages get relayed via the CoHub and for all messages that get relayed, a new message is created.

D. Key Observations and Analysis

HCS scenario is completely random with respect to the mobility paths and destinations, while WDM is somewhat synchronized. With choice of 64 nodes in the simulation, and messages being randomly generated between nodes, it was observed that QoseCo improves fairness by around 4% in many scenarios.

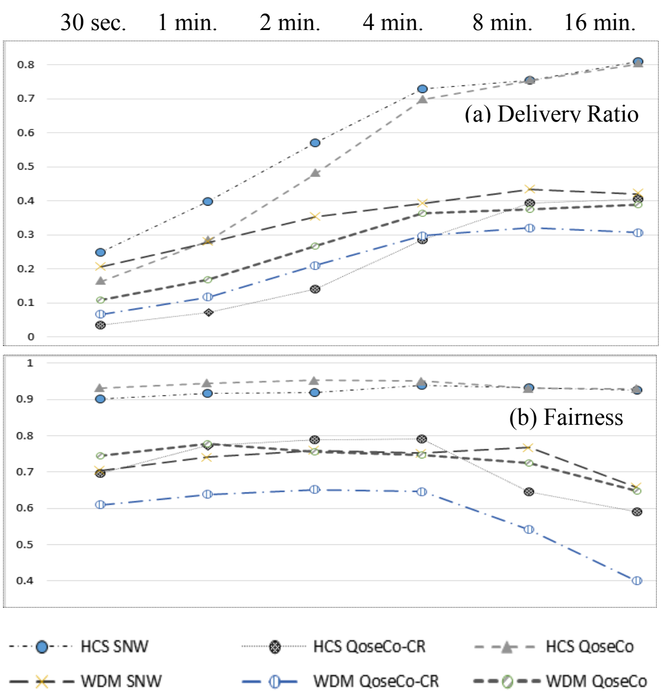

Figure 2 Varying gaps between events

Improvements for fairness in QoseCo are more pronounced when network is congested (frequently generated messages). This happens even as the overheads of QoseCo update messages are present. Poorly serviced nodes are tracked centrally. By moving this computation to central nodes, computation load for other nodes was reduced. The tradeoff was addition of QoseCo update traffic which was less than two percent for most scenarios.

QoseCo-CR is frequently worse off by a factor of about two. For 30 seconds in HCS it degraded beyond a factor of two when compared with QoseCo as severe congestion set in the vicinity of the CoHub.

V. Conclusion and Future Work

The authors proposed a new DTN routing extension and used SNW as underlying routing protocol to demonstrate centrally optimized routing. Using simulations they also showed how centralized routing extension helps improve Quality of Service under congestion. Fairness in delivery ratio was used as the quality metric to demonstrate the impact of QoseCo for real world mobility scenarios. While this work adapts SNW for centralized routing, other DTN routing protocols can also be adapted in similar manner.

This paper also studied the overheads when all messages are relayed through a central node. The cost of relaying the messages via the central nodes is found to be within 2X for most scenarios but degraded significantly as congestion sets in.

The authors are exploring the ability to include more than one location for CoHub(s) and analyze its impact. Moreover present QoseCo implementation does not explore the location and time relationships for the contact. This is another aspect that the team is exploring. Survey of real village to make the working-day-model more realistic is another area that the team is focusing on. The work is also being extended to reduce the number of copies when delivery ratio is high (e.g. above 95%).

References

[1] Fall, K. (2003, August). A delay-tolerant network architecture for challenged internets. In Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications (pp. 27-34). ACM.

[2] Thakur, Abhishek, and Chittaranjan Hota. "Designing an extensible communication platform for rural area." In Advances in Computing, Communications and Informatics (ICACCI, 2014 International Conference on, pp. 1348-1355. IEEE, 2014.

[3] Spyropoulos, Thrasyvoulos, Konstantinos Psounis, and Cauligi S. Raghavendra. "Spray and wait: an efficient routing scheme for intermittently connected mobile networks." In Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking, pp. 252-259. ACM, 2005.

[4] Massri, K., Vernata, A., & Vitaletti, A. (2012, October). Routing protocols for delay tolerant networks: a quantitative evaluation. In Proceedings of the 7th ACM workshop on Performance monitoring and measurement of heterogeneous wireless and wired networks (pp. 107-114).

[5] Cao, Yue, and Zhili Sun. "Routing in delay/disruption tolerant networks: A taxonomy, survey and challenges." Communications Surveys & Tutorials, IEEE 15, no. 2 (2013): 654-677.

[6] Ahmed, Shabbir, and Salil S. Kanhere. "HUBCODE: hub-based forwarding using network coding in delay tolerant networks." Wireless Communications and Mobile Computing 13.9 (2013): 828-846.

[7] Hay, David, and Paolo Giaccone. "Optimal routing and scheduling for deterministic delay tolerant networks." In Wireless On-Demand Network Systems and Services, 2009. WONS 2009. Sixth International Conference on, pp. 27-34. IEEE, 2009.

[8] Sandulescu, Gabriel, Aimaschana Niruntasukrat, and Chalermpol Charnsripinyo. "Guaranteed capacity bounds in intermittently-connected networks: A resource-aware, holistic evaluation." Computer Communications 59 (2015): 12-23.

[9] Pujol, Josep M., Alberto Lopez Toledo, and Pablo Rodriguez. "Fair routing in delay tolerant networks." INFOCOM 2009, IEEE. IEEE, 2009.

[10] Fan, Xiaoguang, Victor OK Li, and Kuang Xu. "Fairness analysis of routing in opportunistic mobile networks." Vehicular Technology, IEEE Transactions on 63.3 (2014): 1282-1295.

[11] Keränen, Ari, Jörg Ott, and Teemu Kärkkäinen. "The ONE simulator for DTN protocol evaluation." In Proceedings of the 2nd international conference on simulation tools and techniques, p. 55. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2009.

[12] Jain, Raj, Arjan Durresi, and Gojko Babic. Throughput fairness index: An explanation. Tech. rep., Department of CIS, The Ohio State University, 1999.