C3: Collective Congestion Control in Multi-Hop Wireless Networks

Mark Shifrin
Faculty of Electrical Engineering
Technion, Israel
Email: shifrin@tx.technion.ac.il

Israel Cidon
Faculty of Electrical Engineering
Technion, Israel
Email: cidon@ee.technion.ac.il

***Abstract*—Networks, including various types of Multi-Hop Wireless Networks (MHWN), such as Wireless Ad-Hoc Network (WAHN) and Wireless Mesh Networks (WMN) ubiquitously possess a user locality property, where adjacent users tend to exchange data with common locally popular destinations. Consequently, current congestion control mechanisms which are based on end-to-end signaling (e.g. TCP) do not make use of the aggregated congestion information and produce many redundant control signals. In this paper, we develop a new Collective Congestion Control (C3) paradigm for MHWNs. We first introduce a composition of the regional user nodes to an hierarchy of groups. Within each group we employ a new Collaborative Token Bucket (CTB) algorithm, which implements an effective distribution of the transmission rates and allowed data reception rates among users for communication with each common destination. Next, we demonstrate by simulations that our method yields a higher throughput and faster convergence to the desired performance level as well as better rate fairness compared to traditional end-to-end congestion control schemes. We show that the C3 implementation results with a much smaller amount of a control traffic which is critical in the MHWN resource limited environment. Finally, we analyze the CTB mechanism and demonstrate its advantages over multiple token buckets solutions.**

I. Introduction

Wireless Mesh Networks (WMN) [15][2][7] and Wireless Ad Hoc Networks (WAHN), [14][28] have been intensively studied in the last decades. We term these networks (WMN and WAHN) collectively as Multi-Hop Wireless Networks (MHWN), as they both use multiple wireless links to forward packets from sources to destinations. One of the most crucial challenges associated with both WAHN and WMN is handling network congestion. This problem arises due the network multi-hop nature, the use of limited and bandwidth variable wireless links between neighbors [34], and the limited scalability of WAHN [24] and WMN [20]. The congestion caused by multiple users sharing the same intermittent links or destinations may cause packets that have already traversed several links and therefore consumed network resources to be lost, and hence reduce the network effective throughput (or "goodput") [6]. Since packet generation patterns are typically bursty, congestion may happen spontaneously in time periods where several nodes transmit simultaneously and over-utilize the network, while at other times, nodes can remain silent and under-utilize network resources. Therefore, congestion control schemes have to react fast to such changing operational periods.

Since the sharing of network paths and destinations is the cause of congestion, the congestion control mechanism needs to dynamically control the traffic injection rates by limiting it at the sources. This is typically done separately and end-to-end, for each source destination pair using positive and negative acknowledgements sent in the opposite direction to the traffic. The most commonly utilized solution is employing the end-to-end TCP feedback mechanism [18] which in turn controls the source rate. However, the additional control traffic in opposite direction on a per-connection basis, causes by itself additional congestion and becomes a burden for the entire network.

There are additional important factors that impair the effectiveness of per-connection congestion control in MHWN. First, radio links have inherent overheads incurred for each packet transmitted between neighbors. Therefore, a large effort has been invested to reduce the amount of control information in MHWNs (in particular WAHN and WMN routing [19] [22][27]). Internet traffic measurements have shown that about 40 percent of the Internet packets are (40 bytes) TCP acknowledgements [36][38][30]. While it is not clear that TCP is the prime protocol choice in a resource limited WAHN [10], the use of any per connection end-to-end congestion control will result in an excessive number of feedback signals that may cause a major increase in network congestion. Second, numerous studies show a ubiquitous locality property possessed by all examined networks, including MHWNs, where adjacent users access common destinations and services. This locality is many times exploited by local cache devices for various services (e.g. DNS, Web, P2P, FTP). It is true for both the Internet and for cellular networks. Among multiple examples we would like to emphasize the research conducted in [21], according to which the miss rate in campus local DNS cache is only about 20%, which demonstrates a high degree of locality for Internet destination lookups. [13] presents web trace statistics collected from home IP service at the local campus which indicates a presence of strong locality of the references. Additional research of locality phenomena at different networks scenarios can be found in [31][1][36]. Adjacent users are likely to access the same local and global services and will share common Internet access gateways. This locality characteristic increases the correlation between the

978-1-4244-6061-8/10/$26.00 ©2010 IEEE

behavior of adjacent nodes and between the type of congestion information they should react to. Therefore, per-connection congestion control causes a major duplication in the feedback traffic. Finally, the need to quickly adjust user transmission and reception rates to changing utilization conditions at different network parts, makes the use of per-connection (potentially a low bandwidth connection) statistics inferior to the use of aggregated statistics that relates to the experience of many packets sent over similar paths to the same destination (or to the vicinity of the destination).

In this paper we present an efficient congestion control mechanism for MHWN that takes advantage of MHWN locality-driven topology structure described above. We first present a partition of the users that share the same destinations to groups (extended to a full hierarchical clustering). The destination (or destination group) aggregates the congestion control information and sends it to the group of users that in turn adjust the traffic rate toward this destination using a collaborative transmission rate control algorithm. Since the user traffic is bi-directional (e.g. upload and download) the users group maintains a collaborative reception rate control as well, using the same hierarchical clustering.

The method of signaling congestion control acknowledgements from a single destination to multiple users was previously proposed in a single hop environment [35]. We extend this idea in a multi-hop topology, taking advantage of locality property and using a collaborative rate control mechanism. Within each user group, we introduce a Collaborative Token Bucket (CTB) algorithm, based on a cooperative mechanism developed in [16] for the context of ATM networks. The distribution of the rates to individual users is coordinated by nodes which are designated for this function. CTB also takes advantage of the native radio broadcast of MHWN to facilitate the rate and token distribution. In addition, we extend CTB to the Hierarchical Collaborative Token Bucket (HCTB) algorithm, by sharing the token generation rates between the neighboring groups. The proposed hierarchical architecture exploits the locality property in order to minimize the control traffic. The aggregated control messages sent from the destination via the user hierarchy, implement in a distributed way the HCTB functionality. This results in both a significant reduction of the total control traffic at the both directions and in a significantly faster convergence of the user traffic to the desired level, as compared with a per-connection congestion control methodology.

To the best of our knowledge this is a first work that takes advantage of the traffic locality to improve the congestion control in MHWN. In addition to the presentation of our algorithms we also develop an analysis of the CTB and HCTB mechanisms and show their advantage over non-collaborative token bucket mechanisms. Finally, our proposed hierarchical structure and algorithms are evaluated through ns2 [40] simulations and are shown to have a much faster convergence to the desired performance and a higher network throughput resulted from minimizing the amount of the control messaging.

The rest of the paper is organized as follows: In the next section we summarize the related work. Section III defines the hierarchical topology, Section IV introduces the CTB and HCTB, and includes their analysis and comparison with non-collaborative algorithms. Section V presents the overall architecture of our congestion control scheme. Finally, ns2 simulation results are presented in Section VI.

II. Related Work

Many works addressed the congestion control problem in MHWN. Most of the previous works addressed important topics that are not covered in this paper. For example, there is a large body of work that proposes improvements to the existing TCP congestion control mechanism or suggest alternative TCP-like (per connection) algorithms specially customized for MHWN. See for example in [23][8][39]. Another example is a support for multipath routing proposed in [26]. This work proposes that packets sent over multiple paths can alleviate the overall TCP performance. Congestion control by hop-by-hop backpressure is discussed in [29][25].

There are also many works that conduct congestion control by groups of nodes sharing certain data. However, none of these works address the specific challenges or take advantage of user locality, and none of these works make use of shared data that is related to common destinations or common sources. Among these works [37] is a congestion control scheme that achieves a spatial spreading of the congestion over a region of nodes using a hop-by-hop congestion control. [11] implements a congestion control mechanism which is based on an accumulation of control messages from neighboring nodes. This facilitates the adaptive computation of the transmission rate. However, it is not based on the congestion incurred at the destinations. A somewhat similar idea of collecting the status of buffers in neighboring nodes is suggested in [3], where the congestion notifications transmitted throughout the transmission path, and in [17], where the control notification of the buffers states are broadcasted by each node. A new protocol called ATP specialized for ad hoc networks is proposed in [33], which may apply for both WAHN and WMN case. The congestion control suggested in this work relies on the feedback information of intermediate nodes traversed by the flows, regardless if these flows have a common destination. In [12] a cooperative game framework is built by collecting and distributing a flow information, however the control-related sharing is within a distance of only one hop. Note that in all the latter examples the sharing of congestion information among adjacent sources is used to repair local congestion and none of these schemes use user and destination collaboration for conducting congestion control over the paths that lead to common destinations taking advantage of the locality property.

III. Hierarchical Clustering

In the following we describe the hierarchical clustering of nodes in the MHWN. For the sake of simplicity, in this paper, we assume a given hierarchical partition, done in accordance to the locality property. We assume that the network topology

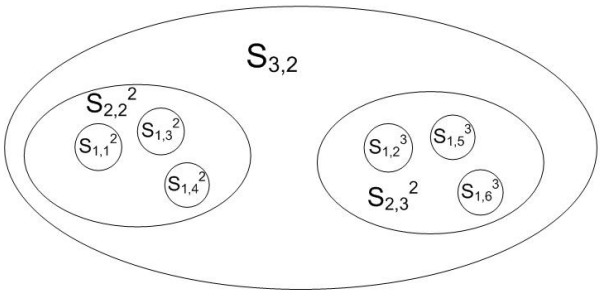

Fig. 1. *Example of hierarchical structure. In this synthetic example* $S_{2,2} = \{S^2_{1,1}, S^2_{1,3}, S^2_{1,4}\}$, $S_{2,3} = \{S^3_{1,2}, S^3_{1,5}, S^3_{1,6}\}$ *and* $S_{3,2} = \{S^2_{2,2}, S^2_{2,3}\}$

is static, or quasi-static, so that the clustering would be stable for relatively long time periods. This assumption clearly holds in the case of WMN, where base-stations are mainly stationary. In the general case, the clustering process can be made dynamic and adaptive to node mobility and locality using known distributed techniques for clustering of mobile nodes [5][9]. Note that in the general case the optimal groups for given destinations overlap, because different subgroups of users can share different destinations. We assume here a simple clustering, where all the groups are disjoint and the clustering is done taking into account the overall level of destination sharing among the users. Note that this assumption also implies that the same clustering can be used for both congestion control, namely regulation of transmission and reception rate of the entire user hierarchy, and for hierarchical routing.

Next, we present a formal definition of the clustering structure that will facilitate the presentation of C3 presented in the next chapters.

A. Formal Definitions

The network is defined by a topology which is decomposed to a hierarchical structure of K levels. Each level consists of multiple groups. Each group is identified with the level it belongs to, and with a unique index at that corresponding level. Let's denote these levels as $k, k = (1...K)$. Denote as n_k the number of groups at hierarchical level k. Note, that in the case $k = 1$ these groups composed of single nodes, otherwise they are groups of nodes, groups of groups of nodes, etc. Denote as i the index of a group at level k. Each group is defined as $S_{k,i} \in T_k, i = (1...n_k)$, where T_k is the set of all groups belonging to the level k, $T_k = \{S_{k,i}\}, |T_k| = n_k$. The groups are indexed arbitrarily and any ordered subset of groups, which belong to the level k, may not necessary be ascribed to the same group of level $k+1$. We term the group $S_{k,i}$ as $S^j_{k,i}$ if $S_{k,i} \in S_{k+1,j}$. The set of all groups at level k, which form group j of level $k+1$ is therefore: $S_{k+1,j} = \{S^j_{k,i}\} | S_{k,i} \subset S_{k+1,j}, i \in [1, n_k]$, where the indexes i, are not necessarily ordered. An example of hierarchical structure is demonstrated in Figure 1. For simplicity, we assume that subgroups of any group do not overlap.

IV. Collaborative Token Bucket

This section presents the Collaborative Token Bucket (CTB). We employ a Collaborative Token Bucket (CTB) among a group of nodes, and further extend it for the group hierarchy. Within a group, several nodes generate traffic to a common destination, while in the opposite direction they receive an incoming data from this destination. Similarly, the nodes can receive data from these destinations that also suffer from the same congestion conditions along similar routes. Therefore it makes sense to control their transmission rates and their reception rates using a collaborative and collective algorithm. CTB introduces a significant advantage for the case of bursty traffic in comparison to non-collaborative rate control. This is because CTB allows active transmitters to take advantage of unused bandwidth of inactive neighbors and to increase their rate till the group maximal rate is reached. Similarly, since the processing or reception ability of each user in the group is limited, it can limit its reception rate through an end-to-end credit system (e.g. TCP reception window). Meanwhile, other neighboring groups can increase their reception rate accordingly, maintaining the total allowed reception rate as constant. Thus, CTB can be effectively operated both for the reception and transmission rate regulation.

A. CTB Principles

In the following we explain the general CTB principles. For simplicity we refer to the transmission rates, but the same discussion valid for the reception rates (or reception window sizes) as well. In order to implement a collaborative token bucket the token bucket control information should be shared among users. Namely, we assume that each subgroup knows the total number of tokens and the number of pending packets in the queues of all the other subgroups in the same group, which stems from the local control messaging (taking advantage of broadcasted messages where available). In addition, there is a distribution of token generation rates, which is performed by a remote congestion control. The detailed explanation of the Collective Congestion Control is described in section V.

CTB is defined by the following structure of indicators and rates:

$E^j_{k,i}$**:** the load indicator of the group $S^j_{k,i}$. The value 1 indicates that the packet queue has packets for transmission and the value 0 indicates that the queue is empty (i.e. the node currently is not transmitting).

$Rb^j_{k,i}$**:** the basic (default) token rate for group $S^j_{k,i}$. This rate is derived by a higher hierarchical level (in case it exists). In the case all the subgroups of the same group are transmitting - this is the actual transmission rate.

$R^j_{k,i}$**:** the actual instantaneous rate of subgroup i at level k, calculated periodically. It depends on the number of active (transmitting) groups at the group j at level $k+1$.

1) Instantaneous Rate Calculation: The actual rate of group i is given as follows ([16]):

$$R^j_{k,i} = \frac{\sum_m Rb^j_{k,m}}{\sum_m Rb^j_{k,m} E^j_{k,m}} \cdot Rb^j_{k,i} \cdot E^j_{k,i} \tag{1}$$

The rate introduced in Equation 1, is the actual current rate. This is the rate in which the tokens are created. This rate is updated periodically. In case the group is inactive its rate is defined as 0, avoiding a division by 0. This implies that if a member of the group is inactive (has no packets to transmit) other members may increase their transmission rates.

2) Analysis of the Collaborative Token Bucket: In order to demonstrate the most basic advantage of the collaborative token bucket we compare two independent token buckets with infinite queue sizes versus a system of two collaborative token buckets. For simplicity we assume that the token pool size is zero, i.e. tokens are not accumulated. Clearly, the same analysis can be used for any token pool size [32]. We assume packet arrivals with Poisson rates λ_1, λ_2 and equally exponentially distributed token generation rate with average μ. In case of two independent buckets the average total number of packets in both queues is given according to the simple $M/M/1$ analysis:

$$N_{ind} = \frac{\lambda_1}{\mu - \lambda_1} + \frac{\lambda_2}{\mu - \lambda_2} \tag{2}$$

Analyzing the 2-Dimensional Markov Chain, the corresponding total average number of packets in CTB is given by:

$$\begin{aligned} N_{coo} &= \frac{\lambda_1 + \lambda_2}{2\mu - \lambda_1 - \lambda_2} \\ &= \frac{\lambda_1}{(\mu - \lambda_1) + (\mu - \lambda_2)} + \frac{\lambda_2}{(\mu - \lambda_2) + (\mu - \lambda_1)} \end{aligned} \tag{3}$$

It is easy to see that $N_{coo} < N_{ind}$. Next, assume that there are n collaborative token buckets with token rates λ_i, and a total token generation rate $n\mu$. In the case all n queues have something to transmit, every bucket generates tokens with average rate μ. Otherwise, the total rate of $n\mu$ is distributed equally for all queues that do have anything to transmit. The average total number of packets is given by:

$$N_{coo} = \frac{\lambda_T}{n\mu - \lambda_T} \tag{4}$$

Where $\lambda_T = \Sigma_{i=1}^{n} \lambda_i$ In case there are n independent token buckets the average total number of packets is given by:

$$N_{ind} = \sum_i \frac{\lambda_i}{\mu - \lambda_i} \tag{5}$$

It is clear, that as the number of collaborative leaky buckets grows, the difference between the total average number of packets in CTB and the separate token buckets becomes much more significant. The stability condition for CTB is $\rho = \lambda_T / n\mu < 1$.

3) Indicator of Queue Occupancy: The original CTB algorithm [16] which uses a discrete indicator makes use of only two different levels of the buffer (e.g., empty and not-empty). It clearly makes sense to extend this to more buffer levels. This is particularly helpful when the sharing of information among the users incurs communication latencies. Denote the total number of packets in the queues of a token buckets belonging to the $S^j_{k,i}$ as $q^j_{k,i}$ and the maximum capacity of the corresponding queues of the subgroups in this group as $qmax^j_{k,i}$ The new definition of the indicator is as follows:

$$E^j_{k,i} = \frac{q^j_{k,i}}{qmax^j_{k,i}} \tag{6}$$

The actual rate of group i is given according to Equation 1, where we substitute the discrete indicator by the expression in Equation 6. Consider a special case where the basic rate is equal for all subgroups in a group (i.e. $Rb^j_{k,i} = Rb^j_k$ holds for all i) and all the subgroups in this group have of the same maximum queue size, denoted by $qmax^j_k$. Then, Equation 1 has the following form:

$$\begin{aligned} R^i_k &= \frac{|P^j_k| \cdot Rb^j_k}{Rb^j_k \cdot \sum_i E^j_{k,i}} \cdot Rb^j_{k,i} \cdot E^j_{k,i} \\ &= \frac{|P^j_k| \cdot qmax^j_k}{\sum_i q^j_{k,i}} \cdot Rb^j_k \frac{q^j_{k,i}}{qmax^j_{k,i}} \\ &= \frac{|P^j_k| \cdot q^j_{k,i}}{\sum_i q^j_{k,i}} \cdot Rb^j_k \end{aligned} \tag{7}$$

Note, that there is a maximum and a minimum value for the rate. The rate cannot be less than the basic rate $Rb^j_{k,i}$, Therefore, the basic rate must be chosen carefully. This basic rate is equal to the actual rate once all the buckets are full, i.e. $q^j_{k,i} = qmax^j_{k,i}$, for all i.

B. HCTB - Hierarchical Collaborative Token Bucket

In order to fully exploit the property of locality, CTB is extended to the hierarchical network clustering. Therefore, we introduce a Hierarchical Collaborative Token Bucket (HCTB). The main idea behind HCTB is updating the default rate of each subgroup according to the load of the other subgroups belonging to the same group. For the sake of simplicity, we assume that the default rate $Rb^j_{k,i}$ is equal for all i in the group. The default (basic) rate therefore is then given by :

$$Rb^j_{k,i} = \frac{Rb_{k+1,j}}{|P^j_k|}, \vee i \tag{8}$$

As we show in appendix A, the default rate is also updated periodically, with a much lower frequency than the actual rate. We can conclude that the actual rate is influenced by both the behavior of the neighboring subgroups in the same groups and by the behavior of the neighboring groups and the subgroups belonging to them.

HCTB can be employed for the transmission rate and reception rate distribution as explained before, and also for the hierarchic distribution of the volume (e.g. aggregated TCP window size) of the incoming traffic. Note that in respect to the

HCTB functioning at the reception direction, each subgroup maintains its instantaneous allowed rate (or allowed quota) according to the local buffering and networking conditions. In the case some subgroups are busy and lower their incoming traffic, other subgroups can take advantage of the extra resources and increase their limit for the incoming traffic, as long as the total predefined limit is sustained.

V. Collective Congestion Control

In this section we introduce a description of the congestion control for hierarchically clustered topology based on locality - the Collective Congestion Control (C3).

A. Basic Implementation

Consider a multi-hop topology, divided into groups and subgroups according to the locality property as presented before. Normally, each group would maintain two local token buckets - one generating tokens for the transmission and another generating tokens for the reception, both regulated by single HCTB mechanism. In order to control the token generation rates inside the groups, a User Group Representative Subgroup (UGRS) is assigned in each group. The local functionality of UGRS is to receive and distribute the states of the queues of the buffers of both of the local buckets of all the members in its group. The UGRS is also responsible for the reception of the corresponding states of all its neighboring groups at the higher level.
The pseudo-code of the implementation of the algorithm run by UGRS is shown in the Appendix A.

B. Transmissions of the Basic Rates by Destinations

In order to complete the presentation of the closed loop collective congestion control, we turn to describe the allocation of the basic rates for the HCTB performed by the users. Without loss of generality we assume that destinations, addressed by a user group may also be adjacent, and can be clustered into groups, similarly to sources. Namely, a group of users may send to groups of adjacent destinations. We term these phenomena a *two-side locality*. Since two-side locality scenario is more general than the simple case of a single destination we explain C3 structure and implementation in a topology under the two-side locality assumption. This description can be easily removed in the case of isolated dispersed destinations. Two-side locality may be valid in certain types of mobile network such as specialized defense or emergency networks where adjacent groups of users (say local emergency units) communicate with other remote groups (command & control units or mobile data-center and application infrastructure).
The main idea is that each group of destinations will transmit the allowed (basic) transmission rate as an aggregated single control signaling stream. This stream may be used for both communication directions, i.e. the allowed total reception rate of the entire user hierarchy will be transmitted to the corresponding group of destinations on this stream as too.

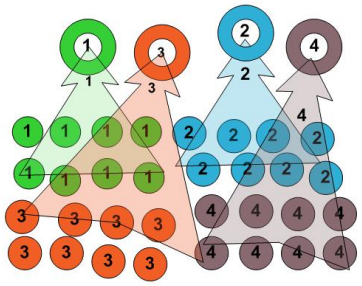

Fig. 2. *Topology 1 - 4 groups of 8 adjacent source nodes. Each group shares a common destination. The four large rings form a group of adjacent destinations. The traffic pattern is that most of the traffic from each source group is directed to its single shared destination*

1) Hierarchical Structure of the Destinations: For the destinations hierarchy, we use the same definitions and structure of the sources hierarchy. Denote the performance level measured at the destination for some source group traffic, described by a value of a performance metric, for example packet loss, as p. Consider some $S_{k,i}$ which transmits to $D_{l,m}$. In case this value represents a common behavior of traffic originated by members within the user group toward a destination group it makes sense to measure the performance (e.g. loss) of the aggregated traffic. A Destination Group Representative Subgroup (DGRS) is assigned, in order to transmit the aggregated packet loss and the traffic intensity values to the DGRS of the upper level group. This DGRS, in turn, will receive the values from all DGRS of its subgroups and will compute the overall performance metric. In the absence of two-side locality each destination is a DGRS. Next, assume that the values of the performance metric were measured over the combined traffic of several source groups by a single destination. Since, the common aggregated loss rate for the entire group is obtained, this lowers the volume of the control messages to be transmitted from the destination to the sources. The decision about the actual number of the control messages and, consequently, the degree of aggregation of the performance statistics of different destinations is a matter of practical optimization. A pseudo-code of our implementation of the corresponding destinations control algorithm is shown in the Appendix A.

VI. Implementation and Simulation Results

This section introduces implementation examples of C3 and HCTB over MHWN using ns2 [40] and presents the corresponding results. We tested the C3 and HCTB with two different topologies, depicted in Figure 2 and Figure 3, and compared convergence rates, throughput and traffic fairness against the same topologies using end-to-end and no congestion control.

A. Convergence

We implemented a hierarchy of three levels at the sources and one level at the destinations, assuming a limited two-side locality (Figure 2). The first level of the sources forms four

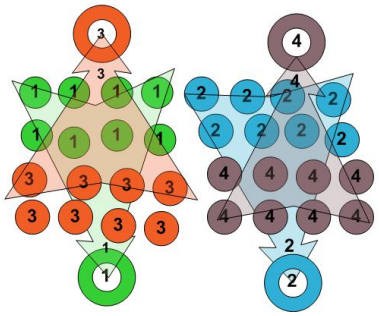

Fig. 3. *Topology 2 - 4 groups of 8 adjacent source nodes. Each group shares a common destination. The four large rings form two destination groups. We used a crossing traffic pattern, which assures all the connections are multi-hop.*

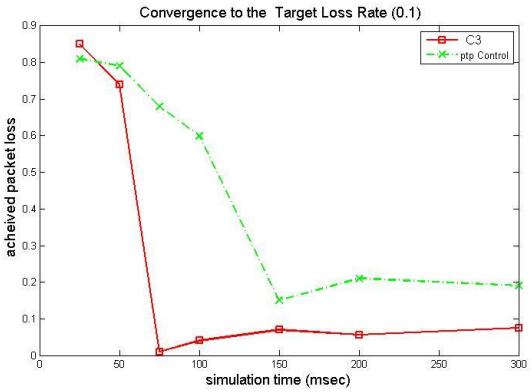

Fig. 4. *Example of convergence for the target packet loss of* $p_{target} = 10\%$

groups of 8 adjacent nodes that send the data to common destinations. Each source node maintains one or more connections to the destination, while each connection handles a collaborative token bucket. The transmission is composed of sequential sessions. The sessions are generated in a random manner where each session consists of several Constant Bit Rate (CBR) packets, and consumes a single token upon a transmission. Each group at the first level has a predefined UGRS and the nodes in this group maintain a connection designated for the control. Each group at the second level is composed of two groups at the first level. One of the UGRS of these two first-level groups functions as a UGRS of the second-level group as well. These second-level groups maintain a control connection with the UGRS at the third level which unites all the sources and maintains a single control connection with the destinations. The UGRS nodes are also responsible for the coordination and signaling of the HCTB control.

The messages transmitted from the destination over the main C3 control connection provide the sources with the ratio for adjusting their basic token generation rate. The ratio is calculated using the following formula:

$$Ratio = 1 + (p_{target} - p_{measured}) \tag{9}$$

This implies that the allowed token generation rate can grow, in case the measured packet loss rate is lower than the target packet loss rate. Otherwise, the current token generation rate is decreased. Each UGRS multiplies the basic rate of it's group by the corresponding adjustment ratio. The result is a new basic rate, adjusted to the latest congestion control feedback. This new rate becomes the base rate for HCTB calculations, till a new ratio value is received. The basic rates of the groups at the lower levels are propagated using the same control connections that serve the HCTB. These rates are calculated according to the formula 8. The destinations maintain a control connection with one (highest level) DGRS and send it the overall calculated packet loss. The comparison of the C3 performance is done against two other schemes that use the same topology and the same offered traffic patterns. In first one, congestion control is done per each connection using a dedicated point-to-point control connection, with no aggregation or collaborations among sources or destinations. The second scheme is not using any congestion control.

One of the important objectives of a congestion control scheme is to quickly converge to the transmission rates at which the target packet loss is reached. During the simulation runtime, in network running the C3, the packet loss was calculated collectively for all the connections within each group of sources, while in point-to-point control case the packet loss was measured for each connection separately (as is typical end-to-end schemes). The convergence to the target packet loss was calculated for the entire network. For this experiment, we used topology 1 for both congestion control schemes. Figure 4 demonstrates a simulation designated to test the convergence speed with topology 1. As expected, it shows a clear advantage to the network that employs C3. Therefore, it is very clear, that an aggregated control channel results in a much faster convergence.

B. Throughput

In order to examine the throughput of a network implementing C3 we used Topology 2, which has a multidirectional data traffic. Such a multi-hop, multi-direction traffic is necessary in order to demonstrate the reduction in network throughput (or "goodput") as the end-to-end offered transmission rate grows. The ratio between the offered load and the throughput was calculated by the two DGRS and sent over two main control connections.

The simulation results of the throughput as a function of the offered source load are presented in Figure 5. In this case, C3 converged to the desired packet loss under any load, while the point-to-point control scheme failed to converge, and resulted with a performance which is even worse than the system that do not use congestion control. The reason for this unexpected poor performance of the per-connection control is the additional congestion resulted from the excessive load inflicted by the control channels.

1) Control Messaging Throughput: We compared the throughput of the control messaging throughout the simulations. The control message loss is significantly lower for C3, which is easily explained by the fact that most of the control messaging is done at a single-hop basis. For example, the

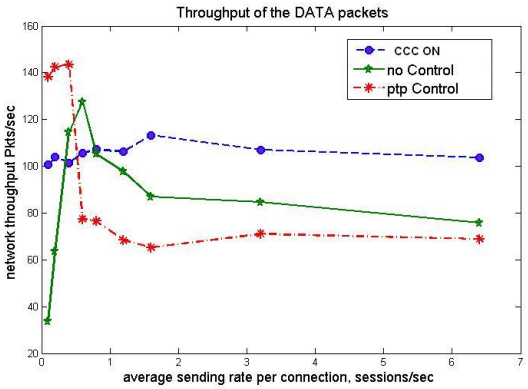

Fig. 5. *Throughput vs. Load Comparison*

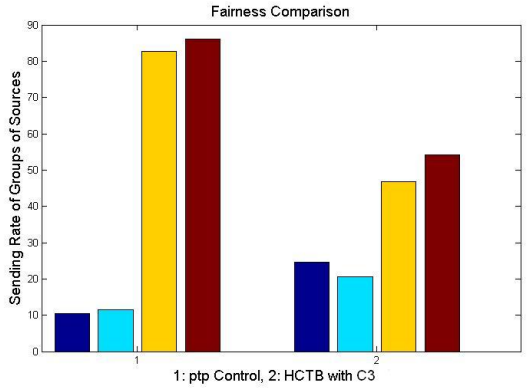

Fig. 6. *Rate Distribution of 4 Groups of Nodes. The 4 columns numbered as 1 present the distribution of the token generation rates during a point-to-point congestion control, while the 4 columns numbered as 2 refer to C3.*

simulation that was designed to converge at a target of 45% data packet loss yielded 10% lower control packet loss. In addition, the total number of control messages for C3 was 15% higher. However, most of the messaging was done between adjacent nodes. This confirms that the multi-hop point-to-point control traffic considerably increases the overall network congestion.

C. Fairness

Naturally, the common control messages allows us to adjust the fairness among the users in a group as desired, without the introduction of additional control messages. In contrast, adding a fairness mechanism to the point-to-point control would imply additional peer-to-peer messaging among the sources or among the destinations and the sources. This would also contradict the end-to-end control paradigm. We summed up the rates of the sources in the point-to-point control topology, which form a groups in C3 topology. The advantage of our topology is clearly demonstrated in Figure 6.

VII. Conclusions

We introduced a new collective congestion control algorithm for MHWN that takes advantage of the important and common property of locality that is ubiquitously observed in most known networks. We defined a locality property where adjacent nodes share common destinations or services. We demonstrated how the locality property can be leveraged for building effective congestion control in MHWN, by aggregating congestion control operations using a decomposition of the MHWN into hierarchical groups rather than using a per-connection control. The source rate control is effectively performed by an Hierarchical Collaborative Token Bucket using inexpensive local control connections and leveraging native MHWN multicast. We demonstrated through analysis and ns2 simulations the significant advantage of our congestion control over a traditional per-connection, end-to-end congestion control. The new scheme results in higher throughput, a faster convergence to the steady state rates and better fairness in rate allocations among nodes.

We believe that the locality property can be further explored for routing mechanisms as well as the combination of routing and congestion control.

Acknowledgments

We thank the anonymous reviewers for their constructive comments.

References

[1] V. Aggarwal, O. Akonjang, A. Feldmann, Improving User and ISP Experience through ISP-aided P2P Locality, IEEE Global Internet Symposium, 2008

[2] I.F. Akyildiz, X. Wang, I. Kiyon, A survey on wireless mesh networks, IEEE Communications Magazine, September 2005

[3] M. Andrews, Joint Optimization of Scheduling and Congestion Control in Communication Networks, Information Sciences and Systems, 40th Annual Conference on, March 2006

[4] Barabási, Albert-László, Scale-Free Networks, Scientific American, May 2003.

[5] S. Basagni, Distributed clustering for ad hoc networks, I-SPAN'99, pp. 310-315, June 1999.

[6] D. P. Bertsekas, R. G. Gallager, Data networks, Prentice Hall, 2nd edition 1992,

[7] E. Bortnikov, I. Cidon and I. Keidar, Scalable Real-time Gateway Assignment in Mobile Mesh Networks, ACM CoNEXT, 2007

[8] K. Chen, Y. Xue, and K. Nahrstedt. On Setting TCP's Congestion Window Limit in Mobile Ad Hoc Networks. ICC '03, May 2003

[9] C. Chiang, H. Wu, W. Liu, and M. Gerla. Routing in clustered multihop, mobile wireless networks with fading channel. IEEE SICON'97, April 1997.

[10] I. Cidon, R. Rom, A. Gupta, and C. Schuba, Hybrid TCP-UDP transport for Web traffic, IEEE Int. Performance, Computing and Communications Conference, 1999.

[11] F. Dressler, Locality Driven Congestion Control in Self-Organizing Wireless Sensor Networks, 3rd International Conference on Pervasive Computing, 2005

[12] Z. Fang and B. Bensaou, Fair Bandwidth Sharing Algorithms based on Game Theory Frameworks for Wireless Ad-hoc Networks, INFOCOM 2004.

[13] S. D. Gribble, E. A. Brewer, System design issues for internet middleware services: deductions from a large client trace, USENIX Symposium on Internet Technologies and Systems, 1997

[14] Z. J. Haas, Wireless ad hoc networks, New York, NY: Institute of Electrical and Electronics Engineers, IEEE journal on selected areas in communications, 1999

[15] G. Held, Wireless mesh networks, Auerbach Publications, 2005

[16] J.S.M. Ho, H. Uzunalioglu, I.F. Akyildiz, Cooperating Leaky Bucket for Average Rate Enforcement of VBR Video Traffic in ATM Networks, Fourteenth Annual Joint Conference of the IEEE Computer and Communication Societies.

[17] B. J. Hogan, M. Barry, S. McGrath, Congestion Avoidance in Source Routed Ad Hoc Networks, 13th IST Mobile and Wireless Communications

[18] V Jacobson, Congestion avoidance and control , ACM SIGCOMM, Association for Computer Machinery 1998
[19] D. Johnson and D. Maltz. Dynamic source routing in adhoc wireless networks, T. Imielinski and H. Korth, editors, Mobile Computing, Kluwer Academic Publishers, 1996.
[20] J. Jun, M.L. Sichitiu, The nominal capacity of wireless mesh networks, IEEE Wireless Communications, 2003
[21] J. Jung, E. Sit, H. Balakrishnan, R. Morris, DNS Performance and the Effectiveness of Caching, ACM SIGCOMM Internet Measurement Workshop, 2001
[22] S. Khurana, N. Gupta, N. Aneja, Reliable Ad-hoc On-demand Distance Vector Routing Protocol, ICNICONSMCL'06
[23] D. Kim, C.-K. Toh, and Y. Choi. TCP-Bus: Improving TCP Performance in Wireless Ad-Hoc Networks. ICC '00, volume 3, June 2000
[24] J. Li, C.Blake, D. S.J. De Couto, H.I. Lee, R. Morris, Capacity of Ad Hoc wireless networks, International Conference on Mobile Computing and Networking, Rome, Italy, 2001
[25] M. Li, D. Agrawal, D. Ganesan, A. Venkataramani, Block-switched Networks: A New Paradigm forWireless Transport, USENIX '09
[26] H. Lim, K. Xu, and M. Gerla. TCP Performance over Multipath Routing in Mobile Ad Hoc Networks, ICC '03, May 2003.
[27] V. D. Park , M. Scott Corson, A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks, INFOCOM '97.
[28] R. Rajaraman, Topology control and routing in ad hoc networks: a survey, ACM SIGACT News, June 2002
[29] B. Scheuermann, C. Lochert, M. Mauve, Implicit Hop-by-Hop Congestion Control in Wireless Multihop Networks, Ad Hoc Networks, 2007
[30] H. Shang and C. E. Wills, Making Better Use of All Those TCP ACK Packets, ISAST Transactions on Communications and Networking, 2007
[31] A. Schmidt, R. Campbel, Internet Protocol Traffic Analysis with Applications for ATM Switch Design, ACM SIGCOMM Computer Communication Review, 1994
[32] M. Sidi, W. Z. Liu, I. Cidon and I. Gopal, Congestion Control Through Input Rate Regulation, IEEE Transactions on Communications, March 1993
[33] K. Sundaresan, V. Anantharaman, H.-Y. Hsieh and R. Sivakumar, ATP: A Reliable Transport Protocol for Ad-hoc Networks, IEEE Transactions on Mobile Computing, November/December 2005
[34] S. Toumpis and D. Toumpakaris, Wireless ad hoc networks and related topologies: applications and research challenges, e & i Elektrotechnik und Informationstechnik, Springer Wien, June 2006
[35] C.-Y. Wan, S. B. Eisenman, A. T. Campbell, CODA: Congestion Detection and Avoidance in Sensor Networks, 1st international conference on Embedded networked sensor systems.
[36] C. Williamson, Internet Traffic Measurement, IEEE Internet Computing, November/December 2001
[37] Y. Yi, S. Shakkottai, Networking, Hop-by-Hop Congestion Control Over a Wireless Multi-Hop Network, IEEE/ACM Transactions on, February 2007
[38] Y. Zhang, J. Luo, H. Hu, Wireless mesh networking: architectures, protocols and standards, Auerbach Publications 2007
[39] J. Zhou, B. Shi, and L. Zou. Improve TCP performance in Ad hoc network by TCP-RC, PIMRC 2003
[40] ns2 Manual
$http://www.isi.edu/nsnam/ns/tutorial/index.html$

APPENDIX

In this appendix we present the HCTB implementation details as well as the different pseudo codes. In our description, all the transmissions are performed using unicast. Note that the availability of broadcast at the first hierarchical level (composed of single nodes) simplifies the implementation in this case.

The implemented algorithm for a HCTB basic rate for some subgroup $S^j_{k,i}$ is as follows:

A. Basic rate calculation algorithm ($Rb^j_{k,i}$):

1) *for each periodic time unit (t_b) do*
   a) *update the basic rate $Rb^j_{k,i}$ according to the last received control transmission from the UGRS of $S_{k+1,j}$*
   b) *for each subgroup $S^i_{k-1,l}$ in $S^j_{k,i}$ do*
      i) *calculate the basic rate $Rb^i_{k-1,l}$ according to Equation 8*
      ii) *if $S^j_{k,i}$ is UGRS, transmit the basic calculated rate $Rb^i_{k-1,l}$ to $S^i_{k-1,l}$*

t_b is the predefined interval of the periodic update for calculating the basic rate.

The implemented algorithm of instantaneous rate of HCTB for some subgroup $S^j_{k,i}$ is as follows (using the queue state indicator):

B. Instantaneous rate calculation algorithm ($R^j_{k,i}$):

1) *for each periodic time unit (t_i) do*
   a) *update the state of the buffer for all $S^j_{k,l}$ in the group, according to the last transmission received from the UGRS*
   b) *transmit to the UGRS the current number of packets in the buffer*
   c) *calculate the instantaneous data rate $R^j_{k,i}$ according to Equation 7*
2) *if $S^j_{k,i}$ is UGRS do*
   a) *update the buffer state for all $S^i_{k-1,l}$ in $S^j_{k,i}$ according to the received control messages*
   b) *transmit to all $S^i_{k-1,l}$ the current number of packets in the buffer*
3) *if $k == 1$ transmit data packets according to the token generation rate $R^j_{k,i}$*

t_i is the predefined interval of the periodic update for the calculation of the instantaneous rate, normally $t_i \ll t_b$. The condition $k == 1$ means that the group $S^j_{k,i}$ is a single node.

The implemented algorithm (for some $D^j_{l,m}$) of the destinations assuming two-side locality is as follows:

C. Destinations control algorithm

1) *for each periodic time unit (t_d) do*
2) *if $l > 1$*
   a) *update the packet loss p, for each subgroup $D^m_{l-1,n}$ in $D^j_{l,m}$*
3) *calculate the aggregated packet loss p and transmit it to the DGRS*
4) *else if $l == 1$ calculate the aggregated packet loss p for all the sources and transmit it to the DGRS*
5) *if $D^j_{l,m}$ has a control channel to the UGRS do*
   a) *calculate the new allowed basic rate for the corresponding UGRS according to Equation 9*
   b) *transmit the new basic rates to the corresponding UGRS, using the corresponding control channel*

t_d is the predefined interval for the periodic transmission of the packet loss updates to the DGRS. The condition $l > 1$ means that this group of destinations has a subgroups and is not a single node.