Quality of Service and Message Aggregation in Delay-Tolerant Sensor Internetworks

Edward J. Birrane III(✉)

Space Department, Johns Hopkins University Applied Physics Laboratory,
Laurel, MD, USA
Edward.Birrane@jhuapl.edu

Abstract. We present traffic-shaping and message-aggregation algorithms that provide reservation-based quality-of-service mechanisms for delay-tolerant internetworks utilizing graph-based routing protocols. We define a Traffic Shaping with Contacts (TSC) method that alters the edge weights in a graph structure to represent service level specifications, rather than physical capacity. This adjustment allows existing routing mechanisms to implement bandwidth reservations without additional processing at the node. We define a Payload Aggregation and Fragmentation (PAF) algorithm that calculates preferred payload sizes over traffic-shaping contacts. PAF aggregates too-small payloads together and fragments too-large payloads to optimize contact capacities. Unlike other mechanisms, TSC/PAF are unaffected by heterogeneous physical, data-link, and transport layer protocols across an internetwork and require only minor modifications to internetwork-layer graph-routing frameworks. Simulation results show that together TSC/PAF reduce the number of messages in a sensor internetwork by 43 % while increasing the goodput of the network by 63 % over standard graph-routing techniques.

Keywords: Delay-tolerant networking · Congestion modeling · Traffic prediction · Quality of service · Fragmentation · Aggregation

1 Introduction

Delay-tolerant networks (DTNs) [1] operate in environments characterized by significant propagation delay, frequent link disruption, and a potentially heterogeneous set of communications hardware. These networks were conceived to enable packetized data communications across interplanetary distances and amongst multiple, heterogeneous spacecraft [2]. DTN architectures also apply to terrestrial internetworks, especially those that include space-based nodes. In terrestrial internetworks, delays and disruptions manifest not only from physical limitations but also from administrative, security, and service policies at local-network boundaries.

Emerging networking architectures, such as delay-tolerant sensor internetworks, provide periodic but deterministically available links. Such architectures use land/space/aerial vehicles to collect data from geographically separated sensor networks (or associated cluster heads). In this paper, we refer to such network architectures as *Challenged Sensor Internetworks* (CSIs) to distinguish them from opportunistically-routed DTNs and other Mobile Ad-Hoc Networks (MANETs).

I. Bisio (Ed.): PSATS 2014, LNICST 148, pp. 58–75, 2016.
DOI: 10.1007/978-3-319-47081-8_6

The periodic and deterministic nature of CSIs allows them to pre-configure future contact opportunities to assist routing mechanisms. A structure for representing contact opportunities between nodes over time is a weighted, directed *Contact Graph* (CG). Given such a graph, paths are computed via any one of a number of graph-theoretic approaches. Since the contact description stored in a CG is generalized, networking functions that restrict their input to the CG operate independent of specific physical, data-link, and transport layer configurations. This is a required feature when considering end-to-end message exchange in CSIs that federate local networks of differing media access, protocols, or administrative privileges.

We investigate a delay-tolerant quality of service (DTQoS) approach for CSIs based on service-level agreements specifying data rates during times of active link connections. These agreements throttle internetwork traffic over individual constituent local networks. Quality of service, in this context, prevents capacity contention between internetwork traffic and pre-existing local network traffic. Our approach specifies how much active link time a constituent local network will devote to the internetwork. We further propose a proactive fragmentation/aggregation scheme to more efficiently bundle user data thus avoiding lower-level protocol fragmentation strategies and their inherent inefficiency.

We define a *Traffic-Shaping Contacts* (TSC) method and a *Payload Aggregation and Fragmentation* (PAF) algorithm to enforce DTQoS agreements and increase the effective use of contact residual capacity, respectively. Together, TSC and PAF provide four benefits in a CSI: (1) payload fragmentation puts small capacities in contacts to useful work, (2) calculation of payload size reduces the chance of downstream fragmentation, (3) aggregation of small data sets reduces messaging overhead, and (4) traffic shaping allows multiple user classes in the internetwork. These benefits can be realized with very little modification to existing CG-based routing schemes.

We analyze the performance of TSC and PAF in an exemplar sensor internetwork simulated using the ns-3 network simulator. Simulation results show these approaches increase network goodput, prevent starvation in networks with multiple classes of data, and allow local networks to enforce the amount of internetwork traffic they are required to carry.

The paper is organized as follows. Section 2 discusses the motivation for this work. Section 3 summarizes related work. Sections 4 and 5 provide overviews and analyses of the TSC and PAF algorithms, respectively. Section 6 presents an ns-3 simulation of these algorithms, demonstrates their performance in reference scenarios, and discusses results. Section 7 concludes the paper.

2 A CSI Architecture

A typical sensor internetwork architecture federates pre-existing, heterogeneous local networks to cover vast or remote regions in a cost-effective manner. We have previously identified three tiers within this architecture: data-generating networks (Tier 1), regionally mobile networks (Tier 2), and exfiltration networks (Tier 3) [3]. The networks in each tier may be separated by distance, by media access, by protocol support,

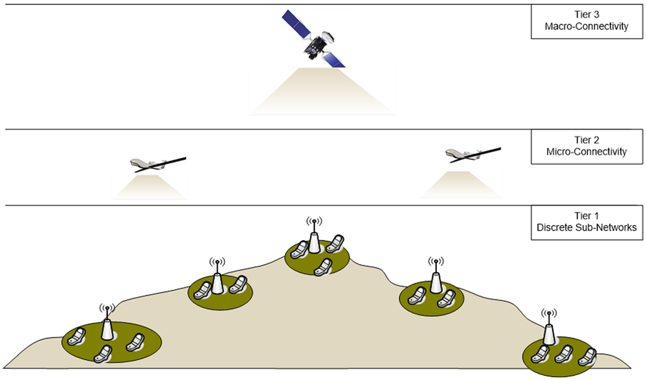

Fig. 1. A federated space-terrestrial sensor web provides applications with link capacity as a function of contact time [3].

or by policy. This architecture is illustrated in Fig. 1 where discrete sensing networks provide data to aerial data mules that forward sensed data to a satellite exfiltration layer.

Constituent networks in the CSI may have pre-existing policies, qualities of service, and classes of users. End-to-end data exchange across the internetwork must work within these existing policies, even though these policies differ in each local network.

Consider the example of three low-powered sensor networks (S1, S2, S3), each with a priority scheme (**H**igh, **M**edium, **L**ow) that dictates how sensed data are communicated from a cluster head to a mobile data mule (such as a jeep driving through each sensor network or an unmanned aerial vehicle) which passes through the three networks in the order S1, S2, S3. The data mule communicates collected data to a Low Earth Orbit (LEO) satellite during a pre-planned pass, with the observation that the short duration of the satellite pass is not sufficient to exchange all data collected. The nominal transmit queue for the data mule is shown in Fig. 2, where received data are kept in a first-come, first-served order. The data mule must have some way to prioritize the data from S1, S2, and S3, but each of these local networks has a different prioritization scheme. If the data mule queues data on a first-come, first-served basis, then the last sensor network reporting data (S3) is given no opportunity to communicate in the upcoming satellite contact.

Figure 2 illustrates two problems that must be solved to negotiate service quality in an internetwork. First, some algorithm must order data in the transmission queue while balancing different local network service quality schemes. Treating the transmission

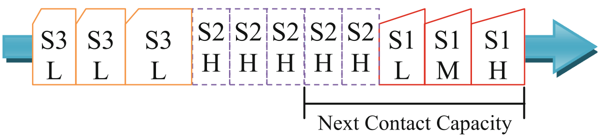

Fig. 2. The transmit queue of an internetworking mobile node must understand individual sub-network priorities to avoid starvation.

queue as First-In, First-Out (FIFO) is clearly unfair as in our example no messages from S3 will be communicated during the next contact. However, attempting to build a sorted transmission queue assumes that some comparison function can be built to consider data from amongst all local networks and produce a linear sorting of transmissions over time. This is not the case when the priority of nodes is dependent upon system time, message age, messages delivered, or any other dynamic quantity. For this reason, classical schemes for packet-based service differentiation do not properly address this issue. Furthermore, even if such a function existed it would grow exponentially as local networks were added to the internetwork.

Second, reducing data volume reduces the loss associated with service differentiation. In Fig. 2, the more data in the transmit queue that can be communicated in the upcoming contact, the fewer data are queued for transmission later. As a sensing internetwork, payloads often share a common destination and therefore there is no advantage to repeating identical header information. The payloads of multiple small messages can be aggregated into a single larger payload to reduce overhead. Conversely, a single large payload may be fragmented into multiple smaller payloads utilizing residual capacities from non-saturated contacts in the CG. Unused portions of the contact may, in aggregate, account for a significant amount of wasted transmission opportunities.

This paper is motivated by the need to (1) practically sort the transmission queue of a mobile node fed by multiple independent networks and (2) balance minimizing messaging overhead and avoiding wasted contact times in bandwidth-constrained sensor internetworks. Mechanisms to accomplish these goals in the context of a CSI are absent in the literature.

3 Related Work

We survey three types of related work: quality of service (QoS) in DTNs, message aggregation in store-and-forward networks, and the use of contact graphs for overlay routing.

3.1 QoS in Heterogeneous Challenged Networks

The term Quality of Service (QoS) refers to a variety of objective and subjective measurements relating to the usability of data received through a network. In the context of any network architecture, a Service-Level Specification (SLS) may objectively define QoS [4]. An SLS, itself, is a set of parameters whose values define the service offered to a traffic stream by a domain [5]. These parameters vary based on the type of network and the protocols and applications using the network. The parameters used to objectively measure QoS in low-latency networks will be different from those used to measure QoS in challenged/disrupted networks.

Research in the area of QoS for heterogeneous networks focuses heavily on selecting between existing QoS services in a given local network (horizontal handoff) or pushing traffic to different networks with other QoS services (vertical handoff) [4].

While the literature is dense with algorithms for horizontal handoffs in support of QoS for low-latency networks, vertical handoffs, which are required in heterogeneous networks, remain an active research area. Approaches to vertical handoffs involve omniscient resource managers and constant end-to-end path monitoring [6] and in general these schemes require sophisticated handover management algorithms, cost functions, and performance evaluation [7]. None of these schemes are appropriate for CSIs with intermittent connectivity as link disruptions prevent the useful collection of information to make these handoff decisions.

In a challenged network end-to-end delivery is not guaranteed, so any QoS mechanism can only be applied at the interface between the application and the network. In addition to best-effort and end-to-end provisioning service classes, a DTN-specific set of service classes focused on allotted bandwidth has been proposed [8]. Allotted bandwidth captures a requested data rate over an interval corrected for link delay, periodic availability, and other latencies. The benefit of this approach is that it informs the queuing limits necessary to communicate a message through a store-and-forward network [8]. Within a given service class there are proposals for priority classes (low, medium, high) and message delivery notifications that could inform scheduling, flow, and congestion control options [9] for the local network.

Consistent with allotted bandwidth service classes, DTN deployments for space networks focus on traffic shaping in accordance with one of four policies: isochronous for real-time data, minimum rate for applications that can tolerate jitter, bulk with deadline for accumulating file transfers, and best-effort [10]. Research on internetworking space networks remains largely focused on dedicated relay nodes and gateways [11].

We discovered no algorithms for the application of these techniques to graph-routed internetworks operating absent service gateways and abstracted from the underlying physical properties of constituent networks.

3.2 Message Aggregation and Fragmentation

Message fragmentation occurs either as a reaction to link congestion or as part of a pro-active flow control mechanism [12]. Reactive fragmentation is typically preferred to pro-active fragmentation in low-latency networks because it results from the instantaneous characteristics of the transmission link, rather than relying on a hard-coded configuration value [13]. In technical demonstrations, both proactive and reactive fragmentation strategies have successfully passed large data across a series of links such that the entire data set could not be communicated over a single link [14]. As the latency of the network increases, the ability to meaningfully measure the state of link congestion decreases making proactive fragmentation a more plausible option. The ability to pro-actively fragment a message based on some network information improves message delivery ratios, as long as unnecessary fragmentation in the network can be avoided [15].

DTN protocols, such as the Bundle Protocol [16], provide mechanisms for fragmentation. These mechanisms typically react to a next-hop bottleneck and are not, alone, appropriate for tuning system-level parameters such as maximizing use of small residual contact capacity. Additionally, these protocols do not implement in-network data fusion, instead relying on users themselves to combine their payload data.

While there is significant literature devoted to reactive fragmentation algorithms, there is very little work on improving proactive fragmentation techniques. Specifically, we have found no work that performs path validation with the purpose of inferring optimal per-path routing fragments. Further, we have not encountered any research performing the inverse function: proactive aggregation at the networking layer when message sizes are small compared to available bandwidth.

3.3 Contact-Graphs for Overlay Routing

Pre-configuring contact opportunities into a CG is a community-supported strategy for planning communications in space networks [17–19] where topological change is deterministic. Contacts in the CG represent the physical bandwidth capacity between two network nodes over time and graph-theoretic algorithms, such as Dijkstra's algorithm [20] and the Contact Graph Routing Protocol (CGR) [21], may be used to build a message path as part of a routing function. CSIs can either preconfigure contacts or infer them from data-link and transport layers of individual links. CSI nodes may synchronize this information over common messaging paths across the internetwork [3]. A concise definition of the CG and its contacts is as follows.

Definition 1. A Contact Graph $CG = (V, E)$ is a directed, weighted graph of contact opportunities, E, amongst nodes, V, in a network, over time. Each node, i, in the network has its own local graph, CG_i. A contact between nodes a,b in a graph is represented by the capacity function $a^b CG_i \in \Re$. Positive values indicate transmission capacity. In the context of a node, we refer to a contact as $C_{a,b} = a^b CG_i$.

Definition 2. The initial transmission capacity (bandwidth) of a contact is measured as the number of bytes that can be received over the contact. Bandwidth (C_{BW}) is initialized as a function of the start and end times (C_{START}, C_{END}), data rate (C_{RATE}), and one-way propagation delay (C_{PROP_DELAY}) of the contact, as follows:

$$C_{BW} = C_{RATE*}(C_{END} - C_{START} + 1 \ - C_{PROP\_DELAY})$$

Some implementations of contact-based routing use the local CG to calculate only the next-hop of a plausible message path [21]. However, an entire message path may be calculated and added to a message as a routing header. This routing header may be evaluated at each waypoint and updated when necessary. In cases where CGs differ amongst nodes in the network, routing headers both prevent loops and enable a wider variety of routing cost functions [22].

4 Quality of Service in CSIs

CSIs cannot provide end-to-end rate-based guarantees as end-to-end connectivity may not exist at any given point in time and active links through the internetwork may have different bandwidths, data rates, and utilization. The concepts of allotted bandwidth, traffic shaping, and bulk delivery in accordance with specified deadlines do remain relevant to CSIs where sensed data need only be available for asynchronous consumption and not part of a real-time or other synchronous data stream.

We make two observations regarding the nature of message delivery within a CSI. First, data sources and sinks are eventually connected by one or more contacts over time. Second, message delivery within a graph-based routing approach is deadline-based: data transmission is only guaranteed to occur such that data can be received prior to the end of the contact. From these observations, we define new terms to describe DTQoS in these networks.

Definition 3. The rate measurement "bits per active second" (*bpas*) identifies the number of bits transmitted from a source to a sink over a DTN/CSI only considering times when links along the path exist. The difference between bits per second (*bps*) and *bpas* is illustrated in Fig. 3. In a CSI, the *bpas* is the amortized bits per second over the duration of the link.

Definition 4. The rate measurement "bits per link" (*bpl*) identifies the number of transmitted bits receivable over a link, assuming no impairments. *Bpl* is less than or equal to the total bandwidth available over a link. It is defined as the *bpas* * the duration of the link, in seconds. For example, a 10 s contact supporting a 30 kbps *bpas* would need to reserve 300 kb of link bandwidth.

Within the context of the CSI architecture, we use a bandwidth allotment mechanism expressed as a *bpas* and enforced on each link as a minimum *bpl*. In other words, so long as there is physical connectivity in the network, DTQoS will provide message delivery in accordance with these new rate measurements. Enforcement becomes a matter of traffic shaping where each node is allowed access to only a portion of the

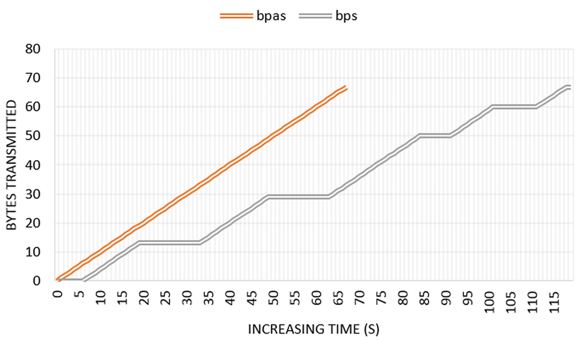

Fig. 3. A performance rate can be defined based on active seconds (left) rather than absolute seconds (right).

physical link. In such a scenario, we can re-draw Fig. 2 as Fig. 4 below. Notably, the *order* in which these messages are transmitted during the contact is irrelevant. The only guarantee is that all of the messages that are queued for the next contact are transmitted prior to the end of the contact.

In this example, *bpl* allocations divide the next link based on configured *bpas* allocation. Specifically to the example in Fig. 4, each of the three tier-1 networks transmit at least one message into the next link whereas in Fig. 2 network S3 would have had no transmission opportunity.

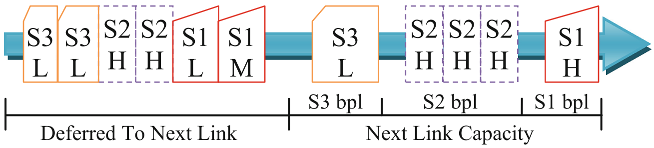

Fig. 4. A prioritized transmission queue ensures that applications using a contact in an internetwork are not starved for bandwidth.

4.1 The Traffic-Shaping Contacts Method

Contacts in a CG are the logical representations of physical links. By slightly altering the definition of a contact, we can represent the *bpl* of the link with respect to a particular user class and SLS rather than the entire link bandwidth. We note that the specification of a particular *bpl* and SLS is negotiated amongst network and internetwork stakeholders as part of network configuration and not automatically calculated on the fly by the network. Algorithms for dynamically altering *bpl* allocations based on local network state is an area for future work.

We augment contact information with a *Unique Identifier* (UID) for a particular SLS such that for the entire bandwidth capacity over a temporal physical link, multiple contacts are defined, one per UID. A reserved UID (*0*) indicates generally available bandwidth, such as that used for retransmission margin or as a reservation pool from which to allocate bandwidth for new UIDs in the future. The relationship between link capacity $\left(L_{cap}\right)$ and contact capacity is captured in Eq. 1, where a and b represent the nodes of a contact, u represents the UID identifying a particular service-level specification, and $C_{a,b,u}$ represents the traffic-shaping contact capturing the allotted bandwidth for user class u over contact $C_{a,b}$.

$$L_{cap} = \sum\nolimits_{u=0}^{u=n} C_{a,b,u} \tag{1}$$

When a message associated with a particular service-level specification is ready for routing at Node i, the set of contacts tagged with the UID for that specification comprises an independent CG, $\mathrm{CG}_{i,u}$, defined in Eq. 2.

$$CG_{i,u} = \left\langle C_{a,b,u} \cup C_{a,b,u0} | a, b \in V \right\rangle \tag{2}$$

In this instance, when a message, m, for the service-level specification identified by u is to be routed in the system, the graph-based routing algorithm at node i uses $CG_{i,u}$ rather than CG_i. In doing so, traffic shaping occurs as a natural consequence of honoring contact capacity. This mechanism does not specify *when* in the course of a contact a particular message will be transmitted, but does guarantee before the end of the contact that there will be a transmission opportunity.

5 Message Fragmentation and Aggregation

In this section, the term message refers to the combination of a user payload and overhead, which we assume is constant, consisting of regular numbers of fixed-length headers. Variable-length headers are of such small size relative to user payloads that we estimate an upper-bound for our constant overhead size without loss of generality.

There is no required correlation between message volume, payload size, and *bpl* allocations. Applications may fill their *bpl* allocations with many fixed-sized messages, with bursts of large messages, or with random message sizes driven by the underlying sensed phenomenology. In all cases, it is unlikely that a set of messages will exactly consume a configured *bpl* allocation; referring back to Fig. 4, we see sections of unused bandwidth within each *bpl* allocation.

For any feasible path through the network, at a given point in time, we can calculate a desirable payload size that balances minimizing wasted bandwidth and minimizing messaging overhead. Nodes in the network calculate this size using information from their local CG and then use this size as the basis for aggregation and fragmentation algorithms. Since contacts in the CG remember their residual capacity, a *Path Preferred Size* (PPS) may be calculated as the smallest residual capacity of any contact comprising the path, less a constant overhead margin, to account for messaging overhead from the physical links and routing headers.

While we desire to keep fragments as large as possible, there is a practical concern relating to the largest fragment allowed in the overlay. Message fragments may be limited by Maximum Transmission Units (MTUs), time-division multiplexing within a contact duration, and duplex operation of a transceiver, all of which may prevent an entire contact opportunity being used for transmission. In such cases, we support a configured maximum payload size which can be lower than the calculated PPS value. Going forward, the term PPS refers to the smaller of this configured size, or the calculated PPS value.

The problem of fitting a series of variable-sized payloads into a calculated PPS is similar to the multidimensional 0/1 Knapsack problem [23], where individual message payloads are the series of knapsack items, $x_1, x_2, \ldots, x_n$, and the knapsack capacity is the overall PPS value. This knapsack problem is known NP-Complete and polynomial approximations remain complex in time and memory. We recommend for embedded systems the use of greedy approximation solutions that first sort messages by value function and then add them to the transmission queue from largest to smallest value [24]. By defining value as the ratio of payload data over *bpl* it is clear that the highest value comes from the largest payload, as it will incur the least amount of messaging overhead. Therefore, finding (or constructing) a payload that exactly matches the PPS represents the optimal use of the current path.

In the remainder of this section we present first an aggregation approach, a message fragmentation approach, and the PAF algorithm that combines them.

5.1 Message Aggregation

If the *PPS* is larger than the current payload size (P_{BYTES}), the sending node has the opportunity to merge several payloads into one aggregate payload (AP*)*. Payloads are candidates for aggregation if they share a destination, UID, associated security mechanism, and any other common attributes that allow a path for one payload to work for all aggregated payloads. The bit savings of a specific grouping of like-payloads, *S*, of the aggregation approach is measured by the *future* bandwidth freed by the aggregation. This is the sum of the contact bandwidth used (and thus not needed for transmitting over future contacts) and the reduction in message overhead, calculated in Eq. 3.

$$S = (AP_{BYTES} - P_{BYTES}) + O * (AP_{MSGS-1}) \tag{3}$$

Where O is the per-message overhead, AP_{BYTES} is the size of the aggregated payload, and AP_{MSGS} is the number of whole payloads aggregated together and thus not requiring per-payload messaging overhead.

When AP_{BYTES} equals PPS we have an optimal result: we have maximized transmission opportunities by using all practical bandwidth along the path and we have minimized message overhead to the overhead of one message. In most cases, however, $AP_{BYTES} < PPS$ and we must decide whether to ignore the delta bandwidth or to fragment a payload to fill this remainder.

5.2 Message Fragmentation

Once the aggregation algorithm has finished, there may still exist some estimated residual capacity (*RC*) left in the AP such that $0 < RC < PPS$. A decision must be made as to whether another payload should be fragmented and used to completely fill the *AP* prior to its transmission.

There are two reasons to fragment a payload at the overlay. First, we avoid wasting capacity as, at the time we perform the calculation, there is no known payload that can fit within *RC*. If no such payload is generated by the application prior to the contact expiration then *RC* bytes of transmission opportunity are wasted. Second, in scenarios where a payload exceeds a given *bpl* allocation, then fragmentation must be performed else the payload will never be transmitted. Conversely, there is a reason to not fragment a payload at the overlay: if the additional overhead burden incurred by fragmentation is greater than the *RC* being saved. This burden increases with the number of fragments generated, because each fragment will incur its own messaging overhead.

Definition 5. The burden, *B*, associated with fragmenting a payload, *P*, is the difference between the total bytes generated from payload fragmentation based on

RC ($\mathrm{Size}_{\mathrm{Cur}}$) and the total bytes generated from payload fragmentation based on the overall *bpl* allocation ($\mathrm{Size}_{\mathrm{Best}}$). When calculating $\mathrm{Size}_{\mathrm{Cur}}$, this first payload fragment (P_{RC}) is made large enough to fill RC (accounting for message overhead) leaving the remaining payload, P-P_{RC}, to be communicated during a future contact.

$$Size_{Best} = \left\lceil \frac{P}{(bpl - O)} \right\rceil * O + P \tag{4}$$

$$P_{RC} = P - (RC - O) \tag{5}$$

$$Size_{Cur} = RC + \left\lceil \frac{P_{RC}}{(bpl - O)} \right\rceil * O + P_{RC} \tag{6}$$

$$B = Size_{Cur} - Size_{Best} \tag{7}$$

When $B >= RC$, we are using as much or more bandwidth than we are saving by fragmentation. For every payload for which $B < RC$, we select the payload whose fragmentation minimizes *B*. We then add (*RC*-*O*) bytes of the payload to AP, and the remaining P_{RC} back into the payload message queue.

When calculating the burden associated with fragmentation we do not directly factor in the link layer overhead. This value is captured as part of the configurable overhead adjustment, *O*.

5.3 The PAF Algorithm

The Payload Aggregation and Fragmentation process is illustrated in Fig. 5. In this figure, the three major processing chains are grouped within boxes and labeled 1, 2, and 3. These processing chains operate as follows:

1. The typical process of passing a message into a routing module to receive the message path is replaced by a probe message. This probe message contains the minimum size of a message (denoted by the message overhead value *O*) and the SLS identifier of the message (the *UID* value). From this path, a PPS value is calculated and the RC of the path is set to this value. Based on the size of the payload versus the RC value the algorithm will aggregate or fragment.
2. The payload is aggregated by including it with the AP. Once this has been done, another payload is retrieved from the payload queue, the RC value is updated to reflect the new AP size, and the algorithm decides whether to further aggregate or fragment.
3. The payload fragmentation burden, *B*, is calculated and used to decide whether the payload should be fragmented or whether it is better to simply waste the contacts RC capacity. If $B < RC$ then fragmentation should be used and the payload fragment is added to the AP and the remainder is put back to the payload queue. In either case, the fragmentation step ends the algorithm and the AP is passed to the transmission queue for that contact.

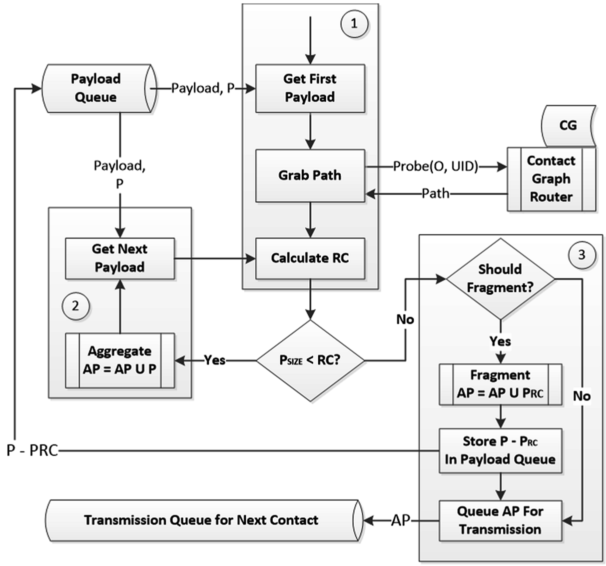

Fig. 5. PAF assembles an aggregate payload from a queue of payloads.

Elided from this algorithm is the more trivial case that if the payload queue is exhausted of payloads, then the AP to date is simply queued for transmission.

6 Simulation and Results

To demonstrate the performance of TSC and PAF in a graph-routed network, we build a representative topology and associated simulation using the ns-3 network simulator. Figure 6 illustrates a federated sensor internetwork in accordance with the architecture presented in Sect. 2 and Fig. 2. Three tier-1 networks (T1.1, T1.2, T1.3) produce data and prioritize them according to local schemes. A tier-2 data collection network consisting of three regionally mobile nodes (T2) provides regular passes through the tier-1 networks to capture data and ferry them to a tier-3 exfiltration node (T3).

The data volume and periodicity of the tier-1 networks are designed to simulate a variety of low-rate monitoring data from sensor cluster heads. T1.1 produces streaming video at 32 kbps, a lower bound for compressed H.264 video. T1.2 produces a bulk file transfer of 1 MB every 7 s. T1.3 produces mixed payloads: a constant 8 Kbps low-priority audio feed, a 45 s half-frame-rate medium priority 16 kbps video feed every minute, and a high priority 4 MB photo every 5 min. These data volumes and production rates simulate the type of competing, variable priority traffic experienced in a federated, resource constrained sensor internetwork. There is no relative priority between T1.1, T1.2, and T1.3; from the point of view of the internetwork, each local network must communicate its data with equal priority.

The T2 mobile nodes follow each other in a circular pattern and take 30 min to complete a circuit. Based on geometry, each sensor network sees a T2 mobile node for

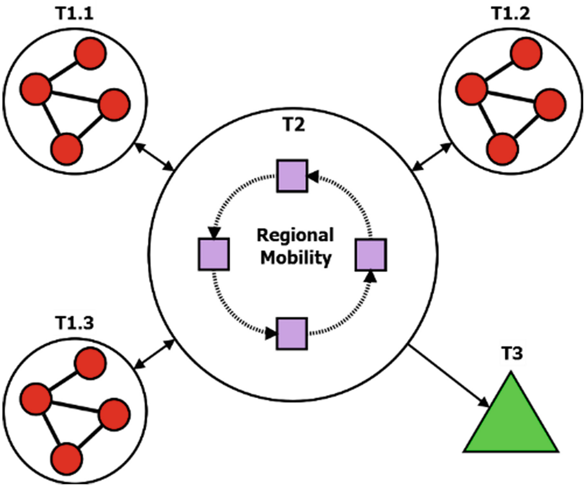

Fig. 6. A multi-tiered ns-3 simulation uses fixed and mobile nodes to simulate a variety of traffic in an internetwork.

approximately 15 out of every 30 min, with each link lasting approximately 5 min. Links in the system are high rate up to 10 Mbps, and we impose a 60 KB limit on the size of any user message payload.

To demonstrate the value of traffic-shaping contacts and message aggregation, we run the simulation in two configurations: one with sufficient *bps/bpl* allotments to account for all user traffic, and again with insufficient allotments. These allotments are given in Table 1. Running the simulation with nominal rates simply results in all data being delivered which is as expected in a network with no challenges for resources. As a constrained network, the regular, small packets produced by T1.1 and T1.3 streaming protocols will need to be managed to preserve bandwidth allocation for the larger, more periodic file transfers from T1.2.

Only the *bpl* allocations are allowed for internetwork data transmission with the remaining bandwidth assumed reserved for other, local network traffic. We run the network with and without TSC/PAF. When running with TSC/PAF, three versions of each contact are defined, labeled with the appropriate T1.1/T1.2/T1.3 UID, and assigned the *"Configured bpl"* from Table 1. Running without TSC/PAF there is a single contact between nodes and no associated UID, and the capacity of the contact is the sum of the three *bpl* values.

Table 1. DTQoS analysis and configuration captures the SLS associated with sensor networks in the TSC/PAF simulation.

Sensor network	bps (Kbps)	Nominal bpas (Kbps)	Nominal bpl (KB)	Configured bpl (KB)
T1.1	32	64	2,400	183
T1.2	1,200	2,400	90,000	5,188
T1.3	136	272	35,116	9,394

For each topological configuration we examine message delivery rate, the ratio of payload bytes sourced verses delivered, and overall bytes delivered.

6.1 Message Delivery Rate

We examine the receipt time of messages at the tier-3 exfiltration node as an indication of how the tier-2 mobile nodes have been populated. As previously stated, the data prioritizations within individual tier-1 networks are not assumed to be comparable at the tier-2 data collection nodes. We expect that the application of the TSC method alters the type and order of message delivery over a delivery with no traffic shaping. Further, we expect that the PAF algorithm will result in fewer messages as payloads are aggregated.

Figure 7 illustrates the message delivery to the tier-3 exfiltration node over time, by originating tier-1 network. Early in the simulation, data are not communicated due to the geometries of the internetwork and the data production. In this figure, message delivery resembles a step function where messages are bulk delivered at the start of contact between the T2 and T3 nodes, as the T2 nodes utilize the full bandwidth of the link without any traffic shaping. Additionally, the T1.1 network, which produces the smallest data volume in the simulation produces and delivers almost half of all messages.

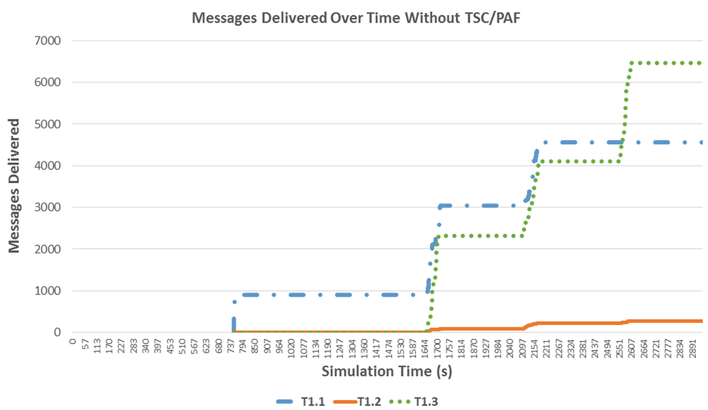

Fig. 7. Without TSC/PAF the T1.1 network, which produces the smallest data volume, dominates message delivery as it sees the data mules first.

Figure 8 illustrates the same data run as Fig. 7, but using the TSC/PAF algorithms. There is a 77 % reduction in the number of T1.1 messages, and a 23 % size reduction in the number of T1.3 messages achieved with using TSC/PAF. The large size reduction for T1.1 messages is based on the ability to aggregate the several small messages that make up its 32 kbps stream. The smaller reduction for the T1.3 messages is due to the periodic 4 Mb file transfer from T1.3 which generates very large payloads that cannot be aggregated. Notably, there is a 200 % size *increase* in the number of messages from T1.2. Without TSC/PAF, the large and periodic T1.2 file messages are overwhelmed by the higher-rate producing T1.1 network. Using TSC/PAF the T1.2 *bpl* allocations are preserved and we are able to receive more of those messages.

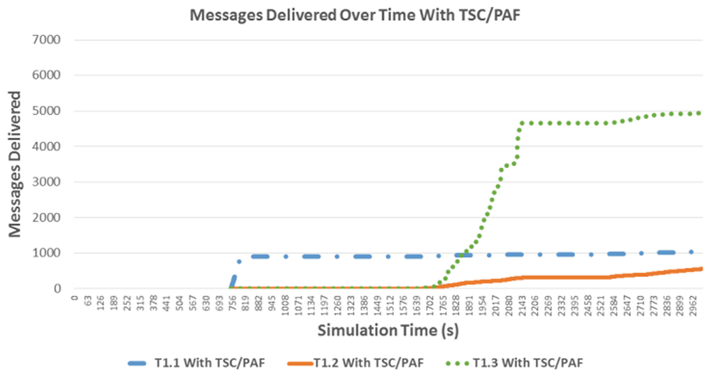

Fig. 8. With TSC/PAF message delivery is even across the three tier-1 networks and accommodates burst bulk file transfers from T1.2.

Figure 8 shows the following three benefits of TSC/PAF. (1) Individual data deliveries are throttled over time as illustrated by smoothed delivery rates (versus the "step function" appearance of Fig. 7) which shows that the internetwork is honoring configured *bpl* values. (2) Approximately twice as many T1.2 data messages are delivered in the system as the TSC method ensures a more equitable sharing of link capacity amongst the three tier-1 networks. (3) The messages from the T1.1 network no longer represent half of all message traffic. As T1.1 represents the smallest data volume, PAF aggregates the high number of small messages into a smaller number of larger messages.

6.2 Payloads vs. Messages Delivered

The number of payloads sent and messages delivered throughout the simulation is shown in Fig. 9. Without TSC/PAF each message contains a single non-aggregated payload and, therefore, the total number of payloads injected into the internetwork (11,299 payloads) is the same as the number of messages delivered by the internetwork (11,299 messages). Several of these messages are small streaming protocol messages where the associated routing and other protocol overhead is a significant percentage of message size. Combining payloads allows PAF to communicate more payload information using fewer messages in the network. With PAF 23,053 payloads are aggregated into just 6,543 messages. This increase in payloads is enabled by the lower overhead associated with having fewer messages and, thus, more capacity to communicate user data.

6.3 Bytes Delivered

Figure 10 illustrates the additional goodput through the internetwork associated with the TSC/PAF algorithms. Without TSC/PAF, a total of 42.39 MB of user data are received through the internetwork, versus 68.92 MB. Our algorithms increased goodput by approximately 63 %. From Fig. 9 we recall that this goodput increase comes with an overall 43 % reduction in the number of messages in the system.

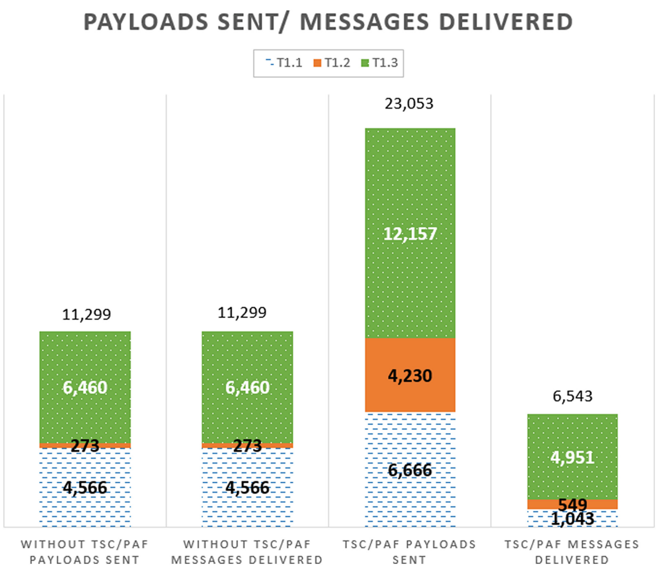

Fig. 9. When using TSC/PAF applications are able to send more payloads in fewer messages.

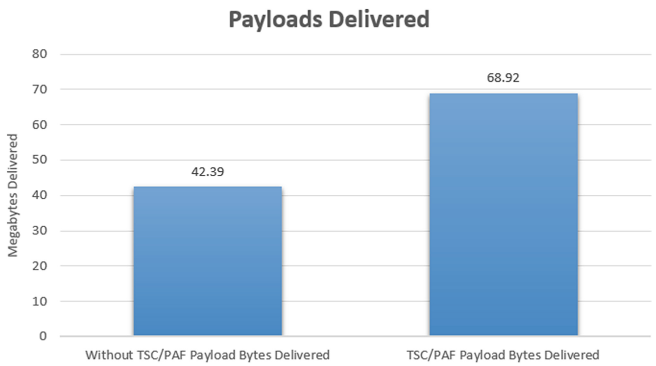

Fig. 10. TSC/PAF efficiencies enable more data to be communicated across the internetwork.

7 Conclusion

Challenged sensor internetworks (CSIs) are supersets of DTNs that support sufficient determinism to build and maintain a CG along commonly used message routes. These internetworks require a minimum quality-of-service provisioning and enforcement mechanism to prevent application data from being passed over for transmission and to limit the amount of internetwork traffic injected into any constituent local network. We present TSC, a method for allowing multiple contacts to share a physical link for the purposes of traffic shaping. In conjunction with TSC, we propose new terms for discussing QoS agreements in CSIs and DTNs, as rate guarantees during periods of link connectivity. We further present PAF, an algorithm for aggregating multiple messages based on preconfigured bandwidth in a CG-routed network. PAF reduces the amount of traffic in a sample sensor network by 43 % while increasing the goodput of the network by 63 %.

References

1. Cerf, V., et al.: Delay-Tolerant Networking Architecture, RFC4838, April 2007
2. Rationale, Scenarios, and Requirements for DTN in Space, Draft Informational Report, CCSDS 734.0-G-0, December 2009
3. Birrane, E.: Building routing overlays in disrupted networks: inferring contacts in challenged sensor internetworks. Int. J. Ad Hoc Ubiquitous Comput. (IJAHUC) **11**(2–3), 139–156 (2012). doi:10.1504/IJAHUC.2012.050271. Special issue on Algorithms and Protocols for Opportunistic and Delay Tolerant Networks
4. Marchese, M.: Quality of Service Over Heterogeneous Networks. Wiley, Chichester (2007)
5. Grossman, D.: New Terminology and Clarifications for Diffserv, RFC3260, April 2002
6. Dugeon, O., et.al.: End to end quality of service over heterogeneous networks: EuQoS. In: Proceedings of NetCon 2005, Lanion, France, November 2005
7. Yan, X., Şekercioğlu, Y.A., Narayanan, S.: A survey of vertical handover decision algorithms in fourth generation heterogeneous wireless networks. Comput. Netw. **54**(11–2), 1848–1863 (2010). doi:10.1016/j.comnet.2010.02.006. ISSN: 1389-1286
8. Demmer, M.: DTNServ: a case for service classes in delay tolerant networks. In: 4th International Conference on Intelligent Computer Communication and Processing, ICCP 2008, pp. 177–184, 28–30 August 2008
9. Caini, C., Cruickshank, H., Farrell, S., Marchese, M.: Delay- and disruption-tolerant networking (DTN): an alternative solution for future satellite networking applications. Proc. IEEE **99**(11), 1980–1997 (2011)
10. Tsao, P., Wang, S.-Y., Gao, J.L.: Space QoS framework over a delay/disruption tolerant network. In: 2010 IEEE Aerospace Conference, pp. 1–5, 6–13 March 2010. doi:10.1109/AERO.2010.5446951
11. Caini, C., Firrincieli, R., Cruickshank, H., Marchese, M.: Satellite communications: from PEPs to DTN. In: 2010 5th Advanced Satellite Multimedia Systems Conference (Asma) and the 11th Signal Processing for Space Communications Workshop (SPSC), pp. 62–67, 13–15 September 2010
12. Fall, K., Farrell, S.: DTN: an architectural retrospective. IEEE J. Sel. Areas Commun. **26**(5), 828–836 (2008)
13. Magaia, N., Pereira, P.R., Casaca, A., Rodrigues, J.J.P.C., Dias, J.A., Isento, J.N., Cervello-Pastor, C., Gallego, J.: Bundles fragmentation in vehicular delay-tolerant networks. In: 2011 7th EURO-NGI Conference on Next Generation Internet (NGI), pp. 1–6, 27–29 June 2011. doi:10.1109/NGI.2011.5985945
14. Ivancic, W.D., Paulsen, P., Stewart, D., Eddy, W., McKim, J., Taylor, J., Lynch, S., Heberle, J., Northam, J., Jackson, C., Wood, L.: Large file transfers from space using multiple ground terminals and delay-tolerant networking. In: 2010 IEEE Global Telecommunications Conference (GLOBECOM 2010), pp. 1–6, 6–10 December 2010. doi:10.1109/GLOCOM.2010.5683304

15. Pitkanen, M., Keranen, A., Ott, J.: Message fragmentation in opportunistic DTNs. In: 2008 International Symposium on a World of Wireless, Mobile and Multimedia Networks, WoWMoM 2008, pp. 1–7, 23–26 June 2008. doi:10.1109/WOWMOM.2008.4594892
16. Burleigh, S., Scott, K.: Bundle Protocol Specification, November 2007. http://tools.ietf.org/html/rfc5050
17. Sekhar, A., et al.: MARVIN: Movement-Aware Routing oVer Interplanetary Networks. In: IEEE SECON (2004)
18. Wyatt, J., et al.: Disruption tolerant networking flight validation experiment on nasa's epoxi mission. In: 2009 First International Conference on Advances in Satellite and Space Communications, pp. 187–196 (2009)
19. Caini, C., Firrincieli, R.: Application of contact graph routing to LEO satellite DTN communications. In: 2012 IEEE International Conference on Communications (ICC), pp. 3301–3305, 10–15 June 2012
20. Segui, J., Jennings, E., Burleigh, S.: Enhancing contact graph routing for delay tolerant space networking. In: IEEE GLOBECOM (2011)
21. Burleigh, S.: Contact Graph Routing: draft- burleigh-dtnrg-cgr-01, July 2010. http://tools.ietf.org/html/draft-burleigh-dtnrg-cgr-01
22. Birrane, E.: Improving graph-based overlay routing in delay tolerant networks. In: Proceedings of IFIP Wireless Days (2011)
23. Fréville, A.: The multidimensional 0–1 knapsack problem: an overview. Eur. J. Oper. Res. **155**(1), 1–21 (2004). doi:10.1016/S0377-2217(03)00274-1. ISSN: 0377-2217
24. Dantzig, G.B.: Discrete-variable extremum problems. Oper. Res. **5**(2), 266–288 (1957)