Communications Quality of Service for Ad-hoc Mobile Optical Free-Space Networks

Ionut Cardei
Florida Atlantic University
icardei@cse.fau.edu

Allalaghatta Pavan
Honeywell Labs

Riccardo Bettati
Texas A&M University

Abstract

Mobile Optical Free Space networks are an emerging technology that will offer low-delay and high speed connectivity using air-borne and ground wireless laser terminals. The time-variable link capacity and dynamic topology make Quality of Service provisioning a difficult problem. We present an architecture for end-to-end statistical delay guarantees. We describe a delay model that uses the concept of virtual traffic to accommodate link capacity variations and transient outages. A mechanism for deploying dependable TCP services is implemented in this QoS-enabled network. The primary-backup TCP replication mechanism is supported by the routing infrastructure and allows transparent server replication. We illustrate performance improvements with simulation results.

1. Introduction[*]

Recent developments in wireless optical technologies in the areas of beam pointing, acquisition and tracking [1] have opened the possibility of building Mobile Optical Free Space (MOFS) networks consisting of ground and airborne terminals carried on airplanes or airships. Long-endurance flight platforms such as NASA's Helios UAV or SansWire's Stratellite, could provide continuous networking coverage via direct wireless laser links and RF links (e.g. WiMAX) to cities or remote communities. Figure 1 illustrates such a network. A multi-hop ad-hoc topology running the IP protocol suite has increased ability to route traffic around areas with adverse weather [2]. At stratospheric altitudes wireless optical links can achieve 2-10 Gbps over hundreds of km. For military applications, MOFS networks can extend Gbps connectivity to remote theaters of operation.

MOFS networks raise significant problems for deployment of applications demanding Quality of Service (QoS). Link quality is time-variable and hard to model theoretically. Signal fading is caused by scintillation from atmospheric turbulence, especially on long links. Absorption from clouds makes links unusable. With multi-aperture nodes alternate links can be set up to form routes that avoid space with adverse transmission effects.

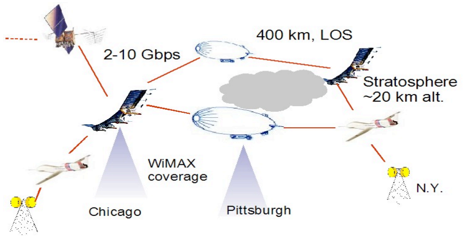

Figure 1. MOFS network concept.

In this paper we present a QoS architecture for MOFS networks, *OptiExpress*, that provides statistical guarantees for delay and fault tolerant TCP services. OptiExpress includes QoS mechanisms (NetEx) capable to accommodate variations in network quality at different levels: link quality variation, short/long term link outages (< 10 ms), topology changes. Our approach leverages standard IP protocols and is implementable on COTS IP routing equipment with little changes. Earlier results related to this project have been published in [3].

QoS frameworks for RF wireless ad-hoc networks have been studied extensively. The main challenge is the variable link capacity and intermittent connectivity. Frameworks for delay guarantees on wired lines ([4]) cannot be applied directly in this context. [5] proposes an approach for statistical delay and drop guarantees in single-hop wireless networks using admission control and earliest deadline first scheduling. Other efforts for QoS in MANETs look into bandwidth reservation and QoS routing. Insignia [6] uses in-band signaling for bandwidth reservation in IP MANETs supporting

[*] This material is based upon work supported by the U.S. Air Force and DARPA under Contract No. F33615-02-C-1247. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the U.S. Air Force and DARPA.

multiple routing protocols.

The next section describes the QoS Architecture. Section 3 presents our approach for QoS-enabled dependable TCP services. Section 4 continues with a review of performance results. We conclude in Section 5 with a summary and comments on future work.

2. QoS Architecture

The OptiExpress QoS architecture gives per class *probabilistic guarantees* for end-to-end packet delays. This means that for any packet that enters the network and is assigned to class *i,* the delay violation probability (DVP) is $P(D^e_i > d^e_i) \leq E_{i.}$, where D^e_i is the random variable for packet delay, d^e_i is the maximum acceptable end-to-end delay, and E_i is the maximum acceptable DVP for class *i* packets. The delays discussed in this paper include only the transmission and queuing delays. The propagation and processing delays can be estimated and factored out.

The end-to-end delay on a route R is: $d_i^e = \sum_{k \in R} d_i^k$, where d^k_i is the maximum delay on link k . The end-to-end delay guarantee is satisfied when

$$P(D_i^e > d_i^e) \leq 1 - \prod_{k \in R} (1 - P(D_i^k > d_i^k)) \leq E_i \ . \qquad (1)$$

One way to compute a partition for end-to-end delay is to split it equally per link: $d_i^k = d_i / |R|$. An alternative is to assign a delay per link inverse proportional to the link capacity: $d_i^k = d_i / (C_k \sum_{j \in R} 1/C_j)$.

To determine the DVP per link, $P(D_i^k > d_i^k)$, the model has two key requirements: 1. network routers use priority packet schedulers, and 2. traffic at any router is policed by leaky bucket shapers, thus traffic arrival random variables are independent. With priority scheduling (highest priority 1), the service curve ([4]) of the total traffic of priority i and higher on a link k is:

$$B^*(t + d_i^k) = \sum_{q=1}^{i-1} \sum_{j \in G_i} B_{q,j}(t + d_i^k) + \sum_{j \in G_i} B_{i,j}(t) \ . \qquad (2)$$

$B_{i,j}$ is the statistical traffic envelope for flow j traffic arrival from flow group $G_{i,}$ (all flows with priority i).

A delay model where the effective capacity for transmission on a link is time-variable *C(t),* is equivalent to a model where the link capacity is constant C, and where a *virtual* highest priority traffic with data rate $C_v(t) = C - C(t)$ models the variable part of the link rate. Figure 2 shows the concept of virtual traffic. Packet delays in the link model with the variable capacity $C(t)$ are equal to delays in the model with the virtual traffic.

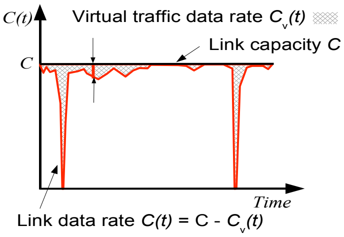
C(t)
Virtual traffic data rate C_v(t) ▨
Link capacity C
C
Time
Link data rate C(t) = C - C_v(t)

Figure 2. Link capacity and virtual traffic.

Define $S(t)$ the stochastic service curve of the wireless optical link. Then, the virtual traffic has a service curve $B'(t) = C\ t - S(t)$. As delay violations occur when effective bandwidth is insufficient, the per-link DVP becomes:

$$P(D_i^k > d_i^k) \leq \max_{t>0} P(B'(t + d_i^k) + B^*(t + d_i^k) \geq C \cdot (t + d_i^k) \qquad (3)$$

Because all the random variables involved are independent, it is possible to compute the end-to-end DVP as in eq. 1 with convolution. We describe next an estimation of B', the virtual traffic service curve.

2.1. Link Estimation

The link capacity C is determined using link parameters, such as data rate and FEC rate. We modeled the virtual traffic that emulates the link capacity time variation with a leaky bucket. Measurements for link faults [7] on wireless optical links indicated link outages of $\tau = 10$ ms that occur on average every T=1.5 s. A simple interpretation for a leaky bucket model is to consider that between outages the bucket fills in with rate ρ_v . When it is full, a link outage "drains" the bucket at link capacity, C.

Thus, we have $\sigma_v = C\tau$ when the bucket drains during outage at full link rate, $\sigma_v = \rho_v\,(T \quad \tau)$ when the bucket fills between two outages, and $\rho_v = C\tau\ /(T \quad \tau)$.

In general, measuring the effective data link capacity *C(t)* may not be practical, as it requires the link to be continuously active at full load. A better approach is to have the link layer record link outages. Let the set of faults be $\{<t_i, \tau_i>\}_{i=1..m}$. t_i is the time when fault i-1 has completed and τ_i is fault i duration. Assuming that prior fault history is a predictor for future behavior, the leaky bucket fault parameters are $\sigma_v = C \max_{i=1..m} \tau_i$ and $\rho_v = \max_{i=1..m} \sigma_v / (t_{i+1} - t_i - \tau_i)$.

2.2. Link Delay Approximation

If the number of flows per link is large enough and flows are independent, the distribution of $B^*(t+d_i)$ can be approximated with the Central Limit Theorem, as in [3], [4]. Let $n_q = |G_q|$, the number of flows in group j on a link. We consider a deterministic leaky bucket arrival envelope for flows in G_j, $b_{i,j}(t) = \sigma_j + \rho_j\ t,$ where ρ_j is the average flow data rate for class j and σ_j is the maximum burst size. By using a Gaussian approximation over intervals, the c.d.f. of $B^*(t+d_i)$ is bounded by a normal distribution $N(\phi_i(t), RV_i(t))$:

$$P(B^*(t + d_i) < x) \leq \Phi\left(\frac{x - \phi_i(t)}{\sqrt{RV_i(t)}}\right) \ , \qquad (4)$$

where $\phi_i(t)$ is the mean aggregate data rate of the flows forming B^*, $\phi_i(t)=(t+d_i)\sum_{q=1}^{i-1} n_q \rho_q + t\, n_i \rho_i$, $RV_i(t)$ is the aggregate rate variance envelope of the B^* flows, $RV_i(t)=(t+d_i)^2 \sum_{q=1}^{i-1} n_q \rho_q \sigma_q + t\, n_i \rho_i \sigma_i$, and Φ is the c.d.f. of the normal distribution.

Before a flow can be admitted in the network, eq. 3 checks whether delay guarantees can be satisfied. This computation involves multiple convolutions and has a high CPU overhead in scenarios with heavy flow arrival rate. To avoid this overhead, the QoS architecture uses Utilization Based Admission Control ([8]), which is insensitive to flow population. All flows from a class *i* share a partition of the link capacity $\alpha_i C$. Admission control limits the total number of flows from a class *i* on a particular link to $n_i = \lfloor \alpha_i C/\rho_i \rfloor$, with $\alpha_i \in (0,1)$ and $\sum_{class\, i} \alpha_i \leq 1$ .This partition guarantees that assumptions for eq. 2 - 4 and DVP limits are satisfied.

The mean rate and the rate variance for the aggregate traffic of the same or higher priority become:

$\phi_i(t)=(t+d_i)\sum_{q=1}^{i-1} \alpha_i C + t \alpha_i C$, and

$RV_i(t)=(t+d_i)^2 \sum_{q=1}^{i-1} \alpha_i \sigma_i C + t \alpha_i \sigma_i C$.

These equations are integrated with eq. 2 - 4 to determine the end-to-end delay guarantees from eq. 1.

2.3. QoS Reconfiguration

As in DiffServ, ingress routers classify IP packets and assigns them to one of *m* traffic classes, according to a service level agreement. A QoS specification for class *i* includes σ_i, ρ_i, the edge-to-edge delay d^e_i, $E_{i,max}$ and the flow priority. The network manager defines the per-class capacity partition $<\alpha_i>_{i=1..m}$. Using routing state and eq. 1, the system determines the DVP limit of the delay along all routes. If for a class *i*, on at least one route, the delay limit is not met, then d^k_i or $E_{i,max}$ could be increased. But this approach is unacceptable for users, as it interferes with application QoS.

The alternative is to reduce the link capacity allocated for real-time traffic classes to a fraction ν, called *safe utilization bound*. The maximum number of flows from a class *i*, admissible on a link *k* of capacity C becomes $n^k_i = \lfloor \nu \alpha_i C/\rho_i \rfloor$, with $\nu \in (0,1)$. ν is computed with a binary search in the (0,1) interval. The selection is based on testing whether the DVP for all classes, on all source-destination routes, falls below the limit $E_{i,max}$. The search stops when ν converges. In our experiments, utilization bounds of 70–90% have been reached. With this approach, admission control for new flows checks whether the maximum number of flows of class *i* has exceeded limit n^k_i, on all links along a route. When routes change, UBAC recalculates ν . After that, UBAC reconsiders admission for all flows for which the source-destination path has changed. Excess class *i* flows are selected for adaptation. Admission is attempted for these flows, in classes with satisfactory QoS, or they are terminated. Adaptation follows the policies configured by the network manager. Adaptation will be discussed in an upcoming paper. The configuration process, followed by re-admission is called QoS Reconfiguration.

2.4. The OptiExpress Architecture

The OptiExpress QoS architecture supports the standard TCP/IP protocols. Routers employ priority-based packet schedulers and leaky bucket shapers to maintain traffic independence and to limit the traffic arrival envelopes. A Bandwidth Broker (BB) service handles QoS reconfiguration (computes ν and n^k_i) and admission control. It maintains data on the per-link aggregate load from real-time flows and keeps the number of flows of class *i* on a link *k* under n^k_i. A configuration protocol transmits classification rules to edge routers. Since the BB needs access to routing state, the QoS architecture requires an interface to the routing protocols. In our simulations the EIGRP routing protocol exported these tables to the BB. The link failure detection of EIGRP was tuned to be sensitive to short failures of 50 ms in order speed up rerouting. Traffic at ingress is classified according to SLAs, policies or negotiated flow admission contracts ([9]). Conforming packets are assigned a Differentiated Services Code Point for priority scheduling. Other packets are set to best effort.

The BB uses TCP as transport protocol. To improve system reliability, an OptiExpress deployment would install the BB service as a replicated TCP service using HydraNet-DS. With this setup, the BB could survive connectivity outages.

Scalability for large MOFS networks (>100 nodes) can be enhanced by partitioning the network topology into QoS Domains. Each domain provides statistical delay guarantees within its boundary.

3. Dependable TCP Services

Service replication is an established method for improving reliability of critical applications in environments where servers and network connections can fail. In the MOFS network, links may experience failures due to weather effects that may not be mitigated quickly enough by rerouting. We present a primary-backup scheme, HydraNet-DS (Direct Server) derived from HydraNet-FT [10]. This system supports deterministic servers. We have integrated HydraNet-DS with the QoS mechanisms presented above.

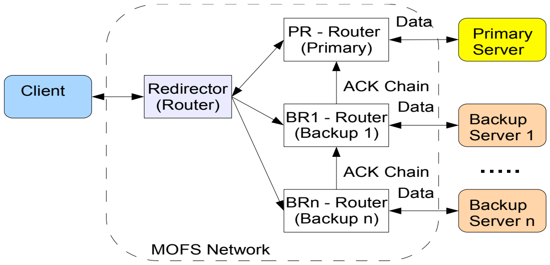

Figure 3. HydraNet-DS architecture.

In HydraNet a primary TCP server has *n* identical replicas distributed in the network. A TCP-based failure detection mechanism allows a backup to quickly take over the primary role. A key feature of HydraNet is that the replication and failover are entirely transparent to client processes/hosts. With our extensions, replication is transparent to the servers as well, requiring no changes to the machines or to the server software. Figure 3 shows a typical HydraNet-DS architecture.

This architecture implements a TCP fault-tolerant service by replicating the server application on multiple machines. All replicas bind to the same TCP port. Clients open one or more TCP connections to the primary server. The service replication architecture implements atomic, ordered, one-to-many from clients to the server group and many-to-one transmission from servers to the clients.

The supporting infrastructure consists of Redirectors (routers connecting clients to the MOFS network), the router connecting the Primary Server to the MOFS network (PR) and routers connecting backup servers to the network (BR_i). A Redirector intercepts packets sent by a client to the primary server. The Redirector has a redirection table describing service access points for each replicated service. When it intercepts a packet from a client, the Redirector forwards it to the primary server and also tunnels it towards the backup server-side routers, $BR_{1..n}$, using an IP-in-IP encapsulation.

To provide ordered and atomic communication, HydraNet-DS builds an ACK chain between the backup server side routers and PR:

$$BR_n \rightarrow BR_{n-1} \rightarrow ... \rightarrow BR_1 \rightarrow PR.$$

All chain routers receive packets from clients, but only BR_n extracts immediately the packet and forwards it to server replica *n*. When BR_n receives a TCP ACK from replica *n*, it forwards only the ACK number to BR_{n-1}, on the ACK chain. Only now can BR_{n-1} forward the client's packet to replica *n-1*. This $delivery_i \rightarrow ACK_i \rightarrow forward_{i-1}$ process propagates down the chain. Router PR eventually delivers the packet to the primary server. This process guarantees that the primary server receives a packet only after confirmed delivery on all other replicas.

Handling server response is similar. When BR_n receives a TCP segment packet from replica *n*, it strips the TCP sequence number and it sends this number to BR_{n-1}. This is repeated up the chain. After BR_i receives the packet with the sequence number for a byte *k* from BR_{i+1}, and also a packet with a byte *k* from replica *i*, BR_i forwards *k*'s sequence number up the chain, towards PR. When PR receives this sequence number, it will forward the corresponding packet to the client after the primary replies. Note that only the packets sent from the primary server will be transmitted to the client. Packets originating from backup servers will not be transmitted on the MOFS network. This protocol makes sure that a client receives a packet only if all replicas have transmitted a corresponding copy.

Message ordering is provided by the TCP sequence number mechanism. Routers involved can recognize TCP retransmissions. If a packet is lost, the TCP flow control will detect a timeout and will retransmit. If a HydraNet router identifies repeated retransmissions, a failure detection protocol is used. If a replica is down or connectivity is lost, the ACK chain will be reconfigured.

Mechanisms for building fault-tolerant TCP services have been proposed before. FT-TCP [11] uses wrappers around TCP server code to forward TCP traffic to a logger that stores the server state. In case the primary fails, a new server will be initialized using the state stored from the previous primary. This system may experience long failover delays due to server initialization. With M-TCP [12] clients open TCP connections to any server from a pool of replicas. Servers log the state of their client connections. Policies control how clients can migrate their connections to another server depending on connection performance. ST-TCP [13] uses a primary-backup scheme, with the backup server intercepting the client-server TCP stream at MAC level. The backup detects primary failures with a heartbeat protocol, allowing rapid failover. Compared to HydraNet-DS, these solutions are less transparent for the server replicas.

4. Performance Results

We have evaluated the performance of the QoS architecture with simulations in OPNET (opnet.com). We present here a set of representative results. We used topologies similar to Figure 1. The network runs Cisco's EIGRP routing protocol, configured to test neighbor reachability every 250 ms. The IP forwarding code has been modified to hold off transmissions to the link layer when the outgoing link is interrupted. The network is loaded with best effort traffic (http and email). The safe utilization bound for CBR real-time flows with 10 ms independent link faults (1.5 s inter-arrival time) was measured at 80.55%.

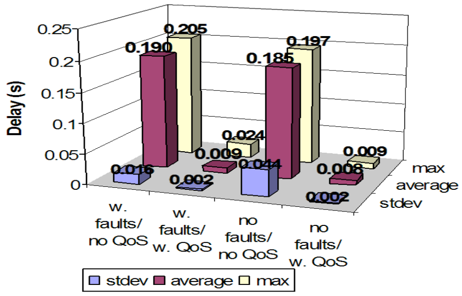

Figure 4. End-to-end packet delay.

The average, maximum end-to-end delays and the standard deviation are shown in Fig. 4 for 4 scenarios combining QoS enabled/disabled, and presence of link faults (10 ms duration/1.5s arrival period). When QoS is disabled, all traffic flows are scheduled FIFO. The reduction in delay is considerable with QoS.

In these experiments we measured a 337 ms QoS reconfiguration, including the EIGRP route convergence time. The aggregate peak overhead was below 1.5% of link capacity.

Table 1 shows TCP performance numbers for an HTTP service configured by HydraNet-DS with one backup server. We measured the average delays for loading a complete 5-object web page, the TCP segment delay and the delay penalty from having the ACK chain. When QoS is enabled, all HTTP flows and the ACK chain will be assigned a higher priority, so the delay drops to a low level. The measured TCP open connection delay penalty is 182 µs and the failover delay is 53 ms.

5. Conclusions

Mobile Optical Free Space networks are an exciting new technology that could be used to provide low-delay and high-speed connectivity to remote networks. The dynamic quality of the wireless optical link hinders deployment of real-time distributed applications demanding end-to-end delay guarantees. In this paper we present a QoS architecture that implements statistical delay guarantees for end-to-end communication in the optical wireless network. In this model statistical service curves represent link capacity and traffic arrival. A model of virtual traffic arrival estimates the capacity variation of the optical links due to atmospheric effects and is integrated with delay violation computation.

We also present an architecture for reliable TCP services based on a primary-backup scheme with transparent deployment and failover for clients and server replicas. Our solution includes in the ACK chain the edge routers connecting the server replicas.

The OptiExpress protocols can be further improved. The centralized bandwidth broker can be replaced with a distributed architecture that uses RSVP and MPLS for enforcing link utilization limits. As an alternative, we consider replicating the bandwidth broker service with HydraNet-DS.

Table 1. HydraNet-DS performance (ms).

Background Load	*HydraNet Configuration*	*Page Delay*	*TCP Delay*	*TCP Segment Delay*	*ACK Chain Delay Penalty*
0% Load	Disabled	1326	301	3.669	0
	Enabled	1348	307	6.350	5.796
100% Load	Disabled	2177	537	28.388	0
	Enabled	2614	703	66.516	25.062
	Enabled, With QoS	1380	310	10.000	11.573

6. References

[1] B. Stadler, G. Duchak, "Terahertz operational reachback (THOR) a mobile free space optical network technology program," in *Proc. of IEEE Aerospace Conference*, 2004

[2] J.Zhuang, M.J. Casey, S.D. Milner, S.A. Gabriel, G. Baecher, "Multi-Objective Optimization Techniques In Topology Control Of Free Space Optical Networks," *Proc. IEEE MILCOM*, Nov. 2004.

[3] S. Wang, R. Nathuji, R. Bettati and W. Zhao, "Providing Statistical delay Guarantees in Wireless Networks," *Proc. IEEE Int. Conf. on Distributed Computing Systems*, 2004

[4] E. Knightly, "Enforceable quality of service guarantees for bursty traffic streams," *Proc. of IEEE Infocom*, 03/1998.

[5] P. Chaporkar, S. Sarkar "Providing stochastic delay guarantees through channel characteristics based resource reservation in wireless network," *Proc. of 5th ACM Int. Workshop on Wireless Mobile Multimedia*, 2002.

[6] S.B. Lee, G.S. Ahn, A.T. Campbell, "Improving UDP and TCP performance in mobile ad hoc networks with INSIGNIA," *IEEE Comm. Magazine*, Vol.39 , June 2001.

[7] Communication from ITT Industries, Oct. 2002.

[8] S. Wang, D. Xuan, R. Bettati, W. Zhao. "Providing absolute differentiated services for real-time applications in static-priority scheduling networks," *Proc. IEEE Infocom'*01.

[9] I. Cardei, R. Jha, M. Cardei, A. Pavan, "Hierarchical Architecture For Real-Time Adaptive Resource Management," *The IFIP/ACM International Conference on Distributed Systems Platforms and Open Distributed Processing*, 04/2000.

[10] G. Shenoy, S.K. Satapati, R. Bettati, "HYDRANET-FT: Network Support for Dependable Services," *The 20th Int. Conf. on Distributed Computing Systems,* Taipei, Apr. 2000.

[11] L. Alvisi, T. Bressoud, A. El-Khashab, K. Marzullo, and Z. Zagorodnov. "Wrapping server-side tcp to mask connection failures," In *Proceedings of Infocom 2001*,

[12] F. Sultan, K. Srinivasan, D. Iyer, and L. Iftode, "Migratory TCP: Connection migration for service continuity over the internet," *Proc. of the 22th IEEE Int. Conference on Distributed Computing Systems*, Vienna, Austria, July 2002.

[13] M. Marwah, S. Mishra, C. Fetzer, "TCP Server Fault Tolerance Using Connection Migration to a Backup Server," *Proc. of IEEE Int. Conf on Dependable Systems and Networks*, San Francisco, CA, 06/2003.