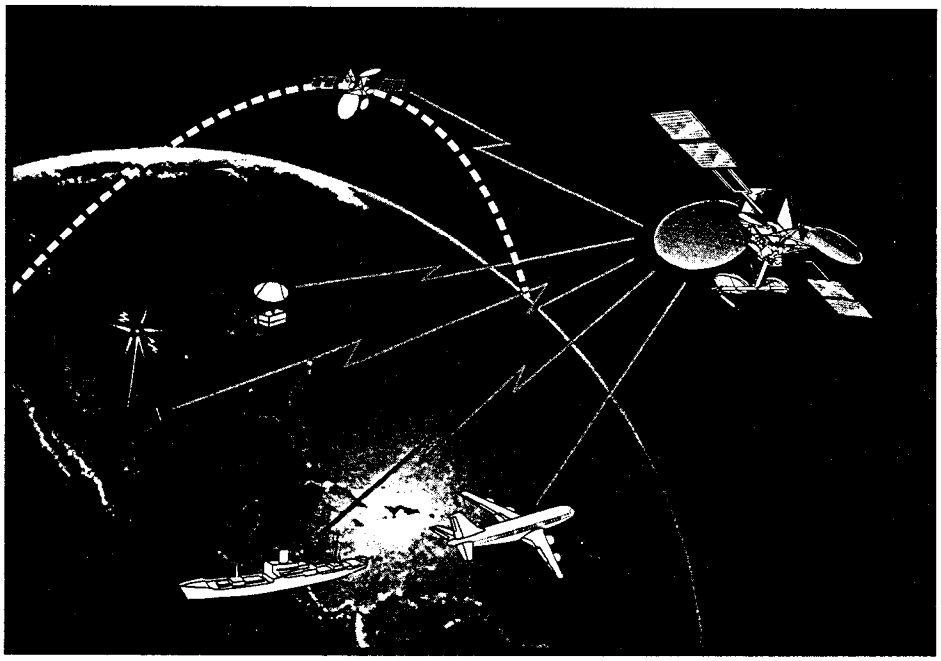
NASA/CP—1998-208524

Satellite Networks: Architectures, Applications, and Technologies



August 1998

Center for Satellite and Hybrid Communication Networks

Flow Control and Dynamic Bandwidth Allocation in DBS-Based Internet

John S. Baras
Center for Satellite and Hybrid
Communication Networks
University Of Maryland
College Park

and

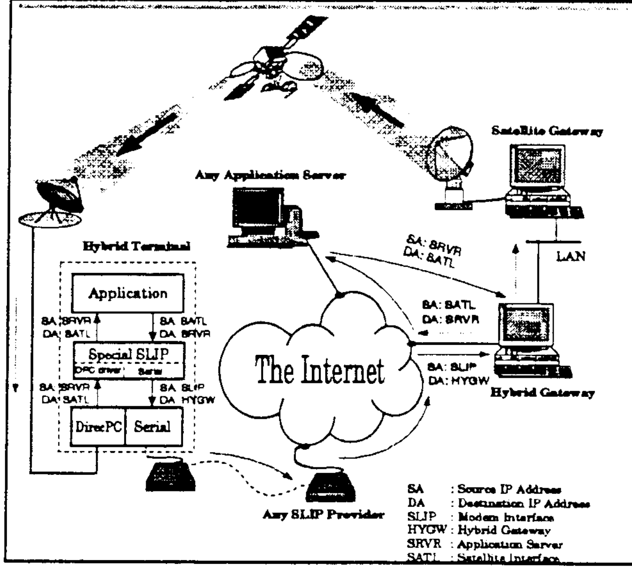
Gabriel Olariu
Hughes Network Systems

Satellite Networks: Architectures, Applications and Technologies
NASA Lewis Research Center
June 3, 1998

DBS - based Hybrid Internet Service

- **Conventional Internet access either too slow or too expensive**
- **DirecPC Turbo Internet™**
  - conceived and designed by the University of Maryland
  - productized and marketed by Hughes Network Systems
- **Awards**
  - 1994 Outstanding Invention of the Year, Univ. of Maryland
  - ComNet '96 New Product Achievement Award (wireless)
  - 1996 "Hot Product", network services, Data Comm. Magazine
  - 1996 Technical Excellence Award (Net. Hardware), PC Mgzine

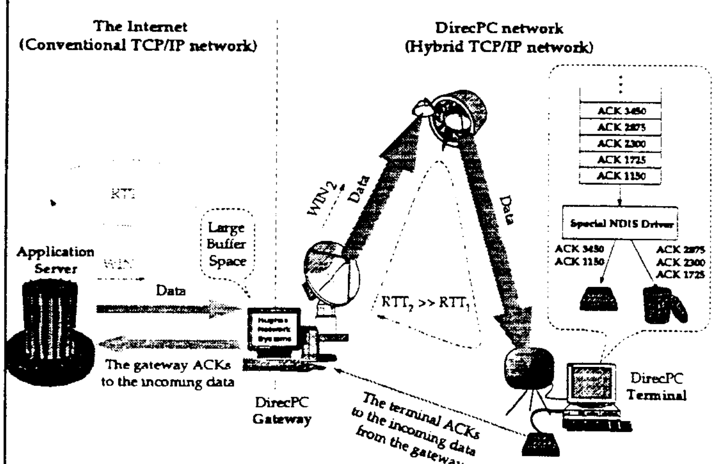
DBS-based Hybrid Internet Service



- Works with any COTS TCP/IP stack, ISPs
- Attached to the Internet through any SLIP
- Traffic to HGW using IP- within -IP encapsulation
- Hybrid Gateway
  - Decapsulates traffic
  - Performs tasks on behalf of HTs for better throughput
- 500 kbps return

DBS-based Hybrid Internet Service: Performance Bottlenecks

A partially observed control problem

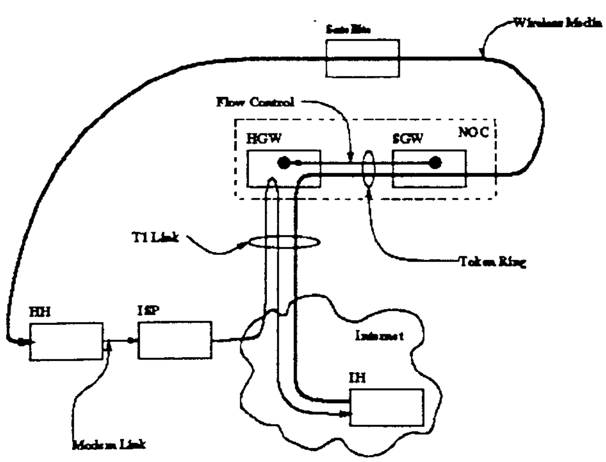


- Problems:
  - TCP over large delay-bandwidth product channels
  - Large No of ACKs
- Solutions:
  - TCP spoofing
  - SACK

Hybrid Internet Service: Extensions

- **Two IETF WGs: TCP over Satellite and Unidirectional routing**
- **Intelligent asymmetric data transmission**
  - Low data-rate (or "short length") via terrestrial
  - High data-rate (or "bulky") via satellite
- **Terrestrial LAN extension of DBS-based Internet**
  - Distribute DBS services from a single receiver to multiple users
  - Satellite hybrid hosts can redistribute data to mobile users
  - "Local loop" anything: Ethernet, ATM, cable TV, wireless
- **Reliable multicast over hybrid networks**
- **Hybrid Internet service over other hybrid network architectures**

Architecture of the Hybrid Internet Service Network

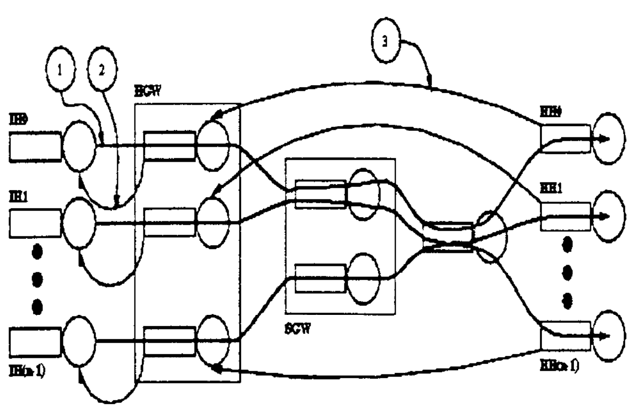


- HH: Hybrid Host
- IH: Internet Host (Server)
- ISP: Internet Service Provider
- HGW: Hybrid Gateway
- SGW: Satellite Gateway
- NOC: Network Operations Center

Network Operations Center (NOC) for Hybrid Internet Service

- **Congestion control :** TCP and TCP Spoofing
  Satellite channel bandwidth allocation
- **HGW : first NOC object that receives data (Router)**
  - HGW prioritizes Hybrid Internet traffic
- **SGW jobs : mixture of Internet and exogenous traffic**
  - Exogenous traffic: package delivery and data feed traffic
  - SGW maintains four queues : two for package delivery and data feed
    two for the two priority levels of Internet
- **Exogenous traffic high priority : fluctuations in bandwidth allocated to Hyhrid Internet**
- **Self-similar traffic: Interactive users as ON-OFF processes**

Flow Control Analysis Model



(1) Data connection:
IS sends data to corresponding HH

(2) Acknowledgments:
From HGW to IS

(3) Acknowledgments:
From HH to HGW

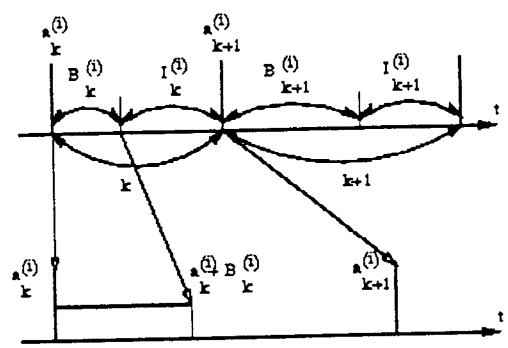
SGW has two queues:
High priority
Low priority

SGW policy: if the number of un-acknowledged bytes for a connection is less than a configurable, but fixed, threshold value, then these packets are high priority

Source Traffic Model

Problem:

- **Independent sources $IS^{(i)}$, i=1, 2, ..., M, send data to HHs via NOC**
- **Find maximum M allowed without producing overflow in the NOC**



$O_k^{(i)} = B_k^{(i)} + I_k^{(i)}$

$B^{(i)}, I^{(i)}$: **Pareto, fin. mean, inf. variance**

Arrival epochs : $a_k^{(i)}$

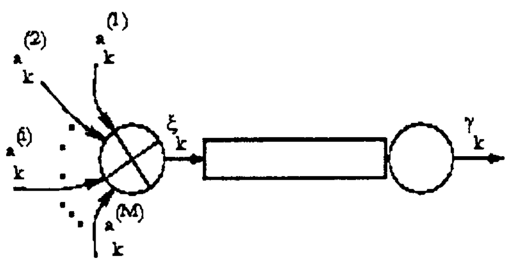
Packet generation rate

$\lambda_k^{(i)} = \begin{cases} \mu_{IS}, & \text{if IS busy} \\ 0, & \text{if IS iddle} \end{cases}$

The Aggregate Process in the Limit of Many Sources

- **Average rate :** $E\left[\lambda_k^{(i)}\right] = \mu_{IS} \dfrac{\mu_B}{\mu_B + \mu_I}$
- **Aggregate arrival traffic: integer valued random point process**
  $$a(M) = \{a_k(M) | k \in Z\}$$
- **Marked point process (Mark = duration of busy period)**
  $$(a(M), B(M)) = \{a_k(M), B_k(M) | k \in Z\}$$
- **Likhanov *et al (1995)*: Take limit as $M \to \infty$, so that** $\lambda = M/(E[B] + E[I]) = const.$, $E[B] = const.$ **and** $E[I] \longrightarrow \infty$
  - $\xi_k(M)$= **No of busy periods arriving at a generic queue at time k**
  - $\xi_k(M)$ **tends to a Poisson with rate**
  - In (a_s, B_s), B_s **is independent from** a_s **and** ξ_s

The Service Facility (NOC)



- Each arrival has service requirement γ_k
- Aggregate traffic shares buffer space
- Source level analysis

- **For individual source we have a G/D/1 queue (constant packet size)**
- **Aggregate traffic is Poisson for large M: So we have a M/G/1 queue**
  - Solve for the stationary state-occupancy probabilities
  - State $X=\{x_{k_i} | k_i \in Z$ = No of sources in the queue at time k_i
  - Arrival process : the aggregate process ξ_{k_i} with rate λ
  - Service process, heavy tailed, Pareto; Stationarity if $\rho = \lambda \mu_B < 1$

The Service Facility (NOC)

- **Probability that *i* new sources will enter queue during one busy period; Used in network dimensioning: An estimate for the No of connections that can be busy during a typical ON period**

$$p_i = \sum_{j=1}^{\infty} P[B=j] \frac{(j\lambda)^i}{i!} e^{-j\lambda}$$

- **Balance equations**

$$q_i = P[X_k = i] \qquad q_0 = 1 - \lambda \mu_B$$

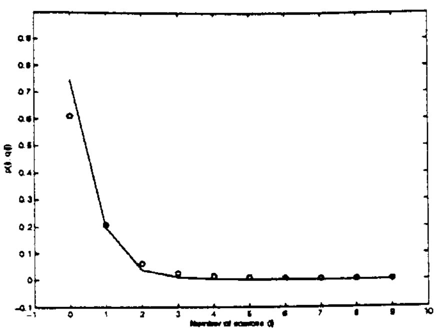
$$q_{j+1} = \frac{1}{P_0}\left[q_j - \sum_{j=1}^{j} p_i \, q_{j-i+1} - p_j \, q_0 \right]$$

- **Packet level analysis: loss probability in finite capacity queue (Likhanov)**

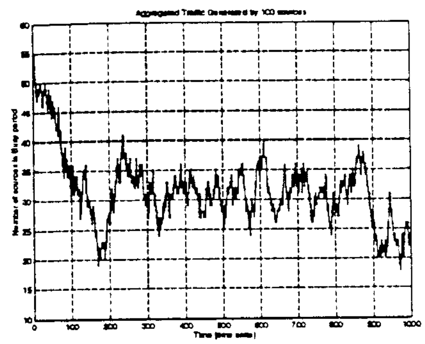
$$P_{loss} \approx \frac{c}{\alpha(\alpha+1)} \lambda^{\alpha} (R\mu_B)^{1+\alpha} L^{1-\alpha}$$

L = buffer length in packets

NOC Simulation Results



Probability that a large number of sources will joint the queue during a busy period
Prob. of No of sources in queue decreases algebraically fast



100 sources aggregated.
Each source: 1 packet / simulation clock
No of sources in busy state at any moment

NOC:
Bandwidth Allocation Strategies

- **All strategies: controller knows (per connection) queue status**
  - Demand at time t : No of packets in queue not sent and unACK, and No of packets that have just arrived
  - Queue length used to determine buffer availability for newly arrived packets
- **Three strategies investigated:**
  - Equal Bandwidth allocation (EB)
  - Fair Bandwidth allocation (FB)
  - Most Delayed Queue Served First Bandwidth allocation (MDQSF)
- **In EB demands may be zero for many instants: waste of BW**
- **FB better for connection requests and min. waste of BW**
- **MDQSF is best**

NOC:
Bandwidth Allocation Strategies

- **Equal Bandwidth Allocation (EB)**
  - Step 1: Find the number of connections with non-zero demand
  - Step 2: Allocate the whole bandwidth equally to connections in the set generated at Step 1
- **Steps 1, 2 performed on-line. Necessitates large computing resources for simulation and for real-world implementation**
- **Demands may be zero for a large set of clock instants**
- **Positive impact on delay, but significant waste of bandwidth**

NOC:
Bandwidth Allocation Strategies

- **Fair Bandwidth Allocation (FB)**
  - Step 1: Find number of connections with non-zero demand
  - Step 2.1: If sum of individual demands ≤ total bandwidth, allocate as requested; END
  - Step 2.2: If sum of individual demands > the resource capacity, go to Step 3
  - Step 3: Divide the total bandwidth to the number of connections in the set generated at Step 1: This generates the ***Fair Share***
  - Step 4.1: For all connections with individual demand ≤ ***Fair Share***, allocate bandwidth to cover the entire individual demand
  - Step 4.2: If cannot perform 4.1, allocate the ***Fair Share*** to all connections
  - Step 5: Find remaining bandwidth after allocating in Step 4.1, go to Step 6
  - Step 6: Re-start from Step 3 with non-zero demand connections for which bandwidth not allocated yet, and the total bandwidth as calculated at Step 5
- **Better than EB in satisfying connection requests and in minimizing the waste of bandwidth**

NOC: Bandwidth Allocation Strategies

- **Most Delayed Queue Served First Bandwidth Allocation (MDQSF)**
  - Step 1: Sort connections in the decreasing order of the delay encountered by the packet in the head of the queue
  - Step 2: Allocate bandwidth starting with the first queue in the ranking generated at Step 1
  - Step 3: Repeat Step 2 until either the entire bandwidth is allocated or, all connections have received service

NOC Simulation Experiments

- **C++ and Matlab environment**
- **Queue model accuracy:**
  - Addition of packets to the queue
  - Keeping copies of unACK messages
  - De-queueing packets
  - Packet delay monitoring
  - Queue length monitoring
- **State: queue length at the service facility**
- **Testing the three strategies:**
  - **Common input data to all strategies**
  - **Test with the same buffer space**
  - **Same total bandwidth**
  - **Same number of sources having**
    - Same succession of ON-OFF periods
    - Same const. arrival rate
- **Service facility has 5 queues, 1 for each connection**
  - Allocation of buffer space to each connection the same
- Packet received service is sent over the satellite channel; a copy is maintained for acknowledgment

NOC Simulation Experiments

- **Following quantities computed, stored and shown graphically**
- **Connection State:** Busy (1) or Iddle (0); All connections use the same constant rate
- **Queue Length (per connection)**
- **Demand:** No of packets admitted in the queue; either new packets or ones that have not received yet service
- **Bandwidth:** No of packets that a queue is allowed to output at a time; It depends on the bandwidth allocation policy; Packets sent to satellite link not deleted from queue until ACKed
- **Delay:** Delay by a packet sent out and not yet ACKed
- **ACKed:** No of packets sent and acknowledged
- **UnACKed:** No of packets sent and un-acknowledged

NOC Simulation Results

- **Comparison of Bandwidth allocation strategies**

Buffer per Connection	500 packets
Total Bandwidth	15 packets/unit time
Number of Connections	5 connections
Constant Arrival Rate	10 packets/unit time
Mean of the Uniform Arrival Rate	5 packets/unit time
Delay Imposed to Queued Packets	0.1 unit time

Common Input Data

Conn1:	1.4469	1.4468	0.0
Conn2:	2.0720	2.0720	0.5298
Conn3:	1.6941	1.6689	0.204
Conn4:	2.0541	2.0524	0.0741
Conn5:	1.7182	1.7088	0.8847
	EB	FB	MDQSF

Average Delays

- **Analytical models and simulation can be used for Network Dimensioning:**
  Estimate No. of sources that can be in the system at the same time