A Design Procedure for Hybrid Circuit-Packet Networks

Martin Krone

BNR INC.
Mountain View, CA. 94043

ABSTRACT

A hybrid network is defined as a certain type of integrated-traffic network in which transmission links are digital and are partitioned into a packet channel and some number of circuit switched channels. An algorithm for designing hybrid interswitch trunk networks is first summarized and then fully specified. The algorithm has been implemented as part of an experimental software system for network planning research. This paper describes methods used for traffic routing, trunk loading, trunk sizing, and facility cost approximation.

1. INTRODUCTION AND BACKGROUND

The description which follows specifies a related set of algorithms for the design of hybrid circuit/packet interswitch networks.

The term hybrid network as used here refers to a network which is designed to integrate circuit switched traffic and packet switched traffic using shared facilities in a particular way. (The term integrated network has often been used in a general way to denote any network which carries mixed traffic types (e.g. voice and data) however implemented.) Specific interest here is in hybrid interswitch networks in which switches at given locations are to be connected by wideband digital trunks. The capacity on each trunk is partitioned into two portions, where one portion is channelized to implement individual circuits and the other is used as a single packet channel. Technologically this would be implemented by time-division multiplexing and allocating a fixed number of bits of each frame per circuit and the remainder to the packet channel [7,11].

The hybrid network design algorithms address one subproblem which arises in designing integrated voice/data networks, and is one component of an experimental software system used for network planning research. Related work [4] describes a set of integrated network scenarios; design problems arising from these are solved by other components of the system.

This procedure draws from the best methods which have become established separately for circuit switched and packet switched network engineering. Unique to this procedure is a way of unifying these that accounts for the economies of shared link facilities and joint routing of the two traffic types. This objective has been aided by descriptions of other hybrid and non-hierarchical network engineering methods [5,6,7]. Circuit switched network engineering methods [1,3,5,10] design networks which meet a blocking performance objective. Links consist of groups of individual circuits, and calls which cannot be set up upon demand are blocked (rejected) in these networks; the methods determine link circuit quantities such that the probability of blocking a call between any pair of end points is less than a given objective value. Since circuit switches generally try alternate paths, the methods have also placed some emphasis on accounting for the resulting traffic overflow among links. Packet switched network engineering methods [8,9] design for a delay objective. Links consist of wideband data channels, and packets which cannot be transmitted upon arrival at intermediate nodes are queued; the methods determine link bandwidths such that average delay between origins and destinations is less than a given objective value.

2. OVERVIEW

2.1 Problem Specification

Switch locations are chosen by preceding design stages and are described by map coordinates or a matrix of interswitch distances.

Traffic loads are specified by separate switch-to-switch traffic matrices for offered circuit switched and packet switched traffic.
Performance objectives are: allowed end-to-end blocking for circuit switched traffic and network average delay for packet switched traffic.

Transmission facilities: parameters (constants) are given for a formula expressing link cost as a continuous function of bandwidth and distance between endpoints, and its derivative. This is a fit to a digital tariff. The actual discrete set of bandwidths available for network construction, corresponding to the tariff, is listed.

Constraints include maximum utilization of packet channels, bandwidth per circuit, and required network connectivity.

2.2 Optimization Method

A network is described by its physical structure and the routing of traffic of each type (these determine trunk loads). The algorithm starts with a trial network design and repeatedly searches for a lower-cost design by redirecting traffic flows toward routes of lower carrying cost, until no further improvement can be found. This method is known as flow deviation [8,9]. The cost associated with an origin to destination route is the incremental cost per unit traffic of loading more traffic on the route, which is mathematically the derivative of network cost with respect to load on the route for the particular traffic type (circuit or packet). For a route it is the sum of incremental costs of links in the route. These will be referred to as link or route pseudolength. They depend upon the loadings on the trial network and are reevaluated each time the routings are changed.

Link bandwidth is modeled as a continuous variable, i.e., the discreteness of the tariff offerings is ignored; but for cost evaluation, and in the final network, links are implemented using the actual available sizes, either the next larger size or several of the next smaller size. This will be referred to as a realizable network.

Packet switched traffic uses a link pseudolength function which mathematically absorbs the sub-problem of minimum-cost bandwidth assignment to links as derived elsewhere [9]. Circuit switched traffic is handled using the Wilkinson equivalent random method to represent the circuit switched load on each link. Then, pseudolength is defined to be the cost per unit marginal traffic capacity, which is the additional traffic which could be carried on one added circuit. For either traffic type, the cost per unit bandwidth is given by the derivative of the bandwidth-cost function with respect to bandwidth.

Circuit switching is represented as non-hierarchical with routing based on the traffic destination. Links adjacent to (outgoing from) each switch are ranked (up to the maximum number of route choices specified for the design) relative to each possible destination switch. The ranking is by pseudolength of switch-to-destination path headed by the links; the link which heads the shortest path is given first rank, etc. It is assumed that switches would attempt to find free circuits for outgoing (originating or tandeming) traffic according to this ranking. This establishes overload relationships.

2.3 Design Variables Determined

Topology. Potentially there can be a link between any switch pair; the procedure specifies which links have non-zero capacity.

Routing. The paths for traffic between each point pair are determined separately for the two traffic types.

Link sizes. Total bandwidth required in each link is determined. Sizes are based upon the routings and resulting link loads of each traffic type.

Realizable design. The design is translated into one that could be realized with the given list of available bandwidths.

3. DETAILED SPECIFICATION

3.1 Notation Conventions

A few general principles have guided notation; understanding these will facilitate reading this section.

Choice of symbols for quantities related to packet switching parallels Kleinrock [9] when possible. Symbols defined there are used here with the same meaning. If a quantity related to circuit switching is analogous to one for packet switching, the preceding alphabet letter is used. Otherwise traditional symbols, if any, are used.

A single subscript of \( l \) denotes a generic link present in the network. Double subscripts refer to point pairs whether or not there is a link.

Upper case is used for matrixes, constants, and sets.

Circumflex ("hat") or primed quantities are non-physical and used for computational purposes.
The following variables and symbols are used in the specification.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A = [A_{ij}]$</td>
<td>Circuit switched offered traffic matrix (Erlangs).</td>
</tr>
<tr>
<td>$P = [P_{ij}]$</td>
<td>Packet switched traffic matrix (kilobits/second).</td>
</tr>
<tr>
<td>$\mu$</td>
<td>Packet length (kilobits).</td>
</tr>
<tr>
<td>$\gamma$</td>
<td>Network-wide total originating packets/second.</td>
</tr>
<tr>
<td>$B$</td>
<td>Blocking objective for circuit traffic.</td>
</tr>
<tr>
<td>$T$</td>
<td>Delay objective for packet traffic (seconds).</td>
</tr>
<tr>
<td>$U$</td>
<td>Packet channel utilization upper limit.</td>
</tr>
<tr>
<td>$W$</td>
<td>Bandwidth per circuit channel.</td>
</tr>
<tr>
<td>$L$</td>
<td>Set of links in the network.</td>
</tr>
<tr>
<td>$C_i$</td>
<td>Allocated bandwidth for packet portion of link.</td>
</tr>
<tr>
<td>$\kappa_i$</td>
<td>Mean circuit load on link $l$ (Erlangs).</td>
</tr>
<tr>
<td>$\nu_l$</td>
<td>Variance of circuit load on link.</td>
</tr>
<tr>
<td>$\delta_l$</td>
<td>Digital load on link (kilobits/sec.).</td>
</tr>
<tr>
<td>$\beta_0, \beta_1, \beta_2$</td>
<td>Bandwidth cost model parameters.</td>
</tr>
<tr>
<td>$\phi_i(C)$</td>
<td>Bandwidth cost function for link.</td>
</tr>
<tr>
<td>$d_{ij}$</td>
<td>Bandwidth cost gradient on link (dollars per kb./sec.).</td>
</tr>
<tr>
<td>$\lambda_i$</td>
<td>Blocking probability on individual link.</td>
</tr>
<tr>
<td>$k$</td>
<td>Pseudolength of link for circuit traffic.</td>
</tr>
<tr>
<td>$l$</td>
<td>Pseudolength of link for packet traffic.</td>
</tr>
<tr>
<td>$K, L$</td>
<td>Matrices of shortest path pseudolengths between node pairs, corresponding to link pseudolengths $k$ and $l$ respectively.</td>
</tr>
<tr>
<td>$n_i$</td>
<td>Number of circuits in link.</td>
</tr>
<tr>
<td>$a_i$</td>
<td>Number of circuits in Wilkinson equivalent random model of link load.</td>
</tr>
<tr>
<td>$\alpha_i$</td>
<td>Offered traffic (Erlangs) in Wilkinson equivalent random model of link load.</td>
</tr>
</tbody>
</table>

Routing for circuit switched traffic: forwarding link choice sequence for routing circuit switched traffic from intermediate or originating node $i$ to destination node $j$. First choice corresponds to shortest path from $i$ to $j$ with link pseudolengths $k$.

Routing for packet switched traffic: corresponds to shortest path with link pseudolengths $l$.

### 3.2 Bandwidth Cost Formula

The cost of bandwidth is modeled by the relationship

$$\phi_{ij}(C) = \beta_0 D_{ij}^2 C^{\beta_2}$$

in which the constants $\beta_0, \beta_1, \beta_2$ have been predetermined to provide a suitably close fit to actual tariffs, and $D_{ij}$ is the tariff-related mileage between nodes $i$ and $j$, and $C$ is a bandwidth (channel rate) in kilobits per second. This expression can be evaluated for any potential link whether or not the link is included in the network topology.

An expression for the derivative of this function is used to evaluate the cost gradients $d_{ij}$ in the progress of the algorithms for links present in the network.

### 3.3 Initial Topology

A network topology is defined by the set of links present in the network. The starting topology is obtained by ranking the potential links by a specific quantity and then incorporating links into the network, preferring them by their ranking, until the network is sufficiently well connected. Connectivity is taken to be represented by minimum node degree.

The link ranking quantity is defined as

$$\max(P_{ij}, P_{ji}) + W(A_{ij} + A_{ji})$$

An alternative way to rank links is to compute, for each link, the cost of the best network in which it is the only backbone link, i.e., all other nodes link to it at an end and not to each other.

### 3.4 Link Blocking Allocation

The design procedure requires that blocking be pre-allocated to links, and then operates with these as constraints independently of how they were allocated. The problem of blocking allocation to minimise cost is, in general, a difficult problem.
In this implementation a current simplification is that a flat value \( b_i \equiv \delta \) is found for link blocking such that the required interswitch blocking \( B \) will be met network wide. The network is modeled as having either one, two, or three link-disjoint paths between each node pair if, respectively, the engineered connectivity is single, double, or greater than double. An appropriate blocking formula is solved for \( \delta \) for each case.

3.5 Flow Deviation

The flow deviation algorithm is the main procedure of the design tool. In the process, traffic loading, link sizing, and pseudolength calculations are used to identify and evaluate the new routes. Flow deviation internally uses a routine for finding all shortest paths (any standard routine can be used) in finding favorable routes, and a circuit switched traffic loading algorithm (see next section). Packet traffic is loaded upon the routes of minimum length; circuit switched traffic is spread among its possible routes by the loading algorithm using link blocking probabilities. The blocking formula in use for circuit switched traffic will be denoted by \( \mathcal{E}(n, \alpha) \) where \( n \) is the number of circuits and \( \alpha \) is the offered random traffic intensity. In this implementation the blocking formula is the Erlang-B formula. Packet traffic is accumulated separately for the two directions on each link; the notation \( \delta_i^{(+)} \) and \( \delta_i^{(-)} \) will be used to denote the two sums. Link sizing is based on the larger of these two sums because actual links are duplex with the same capacity (bandwidth) in each direction.

The definition of bandwidth cost gradient (in Step 6) takes into account the sharing of bandwidth by adding the bandwidth requirements for circuit and packet traffic and evaluating the gradient at the point of total bandwidth. This becomes the value used in the next trial design. The overall result is that routings for each traffic type take into account the economy of scale due to the presence of the other type and its bandwidth.

1. (Initializing.) Assign pseudolengths to links separately for circuit switched and packet switched traffic using pseudorandom positive values for \( k_i \) and \( l_i \), \( i \in L \). Assign

\[
d_i = \beta_i D_i^{\delta_i} \quad (l \in L).
\]

2. Apply the shortest paths algorithm to each traffic type to find path length matrices \( K \) and \( L \) and routings \( g_{ij} \) and \( f_{ij} \) for the current pseudolengths.

3. (Packet traffic loading.) Initially \( \delta_i \leftarrow 0 \) for all links \( i \). For each origin \( i \) and destination \( j \) load traffic \( P_{ij} \) onto each link \( l \) along the shortest route by the appropriate one of:

\[
\delta_i^{(+)} \leftarrow \delta_i^{(+)} + P_{ij} \\
\delta_i^{(-)} \leftarrow \delta_i^{(-)} + P_{ij}.
\]

Total link loads are

\[
\delta_i \leftarrow \max(\delta_i^{(+)}, \delta_i^{(-)}) \quad (l \in L).
\]

4. (Circuit traffic loading.) Use the circuit switched traffic loading algorithm (see next section).

5. (Sizing and bandwidth of each link.) For packet traffic:

\[
C_i = \delta_i \left( 1 + \frac{1}{\gamma T} \sum_{i \in L} \sqrt{\delta_i^+ d_i} \right)
\]

For circuit traffic, apply Wilkinson equivalent method with Rapp approximation of load model:

\[
z = \frac{\nu_i}{\kappa_i} \\
\tilde{\alpha}_i = \tilde{\alpha}_i + 3z(z - 1) \\
\tilde{n}_i = \frac{\kappa_i + z}{\kappa_i + z - 1} - \kappa_i - 1
\]

and then determine link circuit sizing as the least number of circuits \( n_i \) such that

\[
\mathcal{E}(n_i + \tilde{n}_i, \tilde{\alpha}_i) \leq b_i.
\]

The total bandwidth requirement of link \( l \) is

\[
\frac{C_i}{U} + W n_i.
\]

6. Update pseudolengths and bandwidth cost gradient for further iterations.

\[
k_i = \frac{W d_i}{\tilde{\alpha}_i [\mathcal{E}(n_i + \tilde{n}_i, \tilde{\alpha}_i) - \mathcal{E}(n_i + n_i + 1, \tilde{\alpha}_i)]]} \\
l_i \leftarrow d_i \left( 1 + \frac{1}{\gamma T} \sum_{i \in L} \sqrt{\delta_i^+ d_i} \right) \\
\phi_i \leftarrow \frac{C_i}{U} + W n_i.
\]
7. Find a corresponding realizable network and its cost. If the cost is less than that of the previous design, save this as the new design and return to Step 2. If not, terminate and return this design as the solution.

3.6 Circuit Switched Traffic Loading

Circuit switched traffic is to be loaded onto the network in accordance with the manner in which the switches would route calls. Traffic routed through each switch is loaded upon multiple forwarding link choices, each of which blocks some traffic. This spreads traffic among route choices.

The algorithm hands forward parcels of traffic from origins toward destinations, keeping track of traffic to be routed further (separately per destination), and accumulating link loads. Partially routed traffic is denoted below by primed symbols. We will refer to a node as vacated if it has no partially routed traffic remaining to be forwarded. The procedure terminates when all nodes are vacated.

The following simplifications are incorporated.

- For any parcel of traffic offered to a link, the variance of the corresponding overflow parcel is calculated from the allocated link blocking and the offered parcel intensity, without regard for any other parcels loading the link.
- Parcels which have overflowed more than once are treated as if they had overflowed only once from a link with a composite blocking equal to the product of the individual link blockings.

For notational simplicity, in steps 2 and 3, once a particular origin-destination pair \( i, j \) has been selected the subscripts for \( i \) and \( j \) will be omitted.

The classical formula for variance of overflow traffic will be denoted as

\[
V(b, A) = bA \left( 1 - bA + \frac{A}{n_0 + 1 - A + Ab} \right)
\]

for a link with offered traffic \( A \) and blocking \( b \), in which the corresponding (fictitious) number of links \( n_0 \) which would match this blocking and offered load is first solved for such that

\[
\epsilon(n_0, A) \leq b < \epsilon(n_0 - 1, A).
\]

1. (Initializing.) Mark all nodes unvacated. Set all loads to zero, and

\[
A'_{ij} \leftarrow A_{ij}
\]

for all nodes \( i \) and \( j \).

2. Select an unvacated node \( i \) and destination node \( j \) with traffic to be forwarded (positive \( A'_{ij} \)). Identify \( \nu \) route choices for this \( i, j \) by finding the sequence of outgoing links (and corresponding neighbor nodes) which head paths with increasing pseudolength to \( j \):

\[
k_f(v) + K_{m_{s,j}} \leq k_f(v) + K_{m_{s,j}} \leq \cdots \leq k_f(v) + K_{m_{s,j}}.
\]

3. (Traffic hand-forward.) Add offered load intensities to links.

\[
\kappa^{(1)} \leftarrow \kappa^{(1)} + A'
\]

\[
\kappa^{(2)} \leftarrow \kappa^{(2)} + b^{(1)} A'
\]

\[\vdots\]

\[
\kappa^{(v)} \leftarrow \kappa^{(v)} + b^{(1)} b^{(2)} \cdots b^{(v-1)} A'.
\]

Add offered load variance to links.

\[
v^{(1)} \leftarrow v^{(1)} + A'
\]

\[
v^{(2)} \leftarrow v^{(2)} + V(b^{(1)}, A')
\]

\[\vdots\]

\[
v^{(v)} \leftarrow v^{(v)} + V(b^{(1)} b^{(2)} \cdots b^{(v-1)}, A').
\]

Mark any receiving neighbor nodes as vacated. Add handed traffic parcels to their partially routed traffic; remove all traffic from the sending node.

\[
A'_{m_{s,j}} \leftarrow A'_{m_{s,j}} + (1 - b^{(1)}) A'
\]

\[
A'_{m_{s,j}} \leftarrow A'_{m_{s,j}} + b^{(1)} (1 - b^{(2)}) A'
\]

\[\vdots\]

\[
A'_{m_{s,j}} \leftarrow A'_{m_{s,j}} + b^{(1)} b^{(2)} \cdots (1 - b^{(v)}) A'
\]

\[
A'_{i,j} \leftarrow 0.
\]

4. Mark the sending node vacated if it has handed over all traffic for all destinations. If there are any unvacated nodes, return to step 2. Otherwise, terminate.
4. CONCLUSION

A method for the design and optimization of hybrid shared-bandwidth networks has been established. The sharing of bandwidth has been taken into account by establishing a suitable definition for the incremental cost of routing additional traffic on links. Formulas from the literature and engineering practices have been applied to size circuit groups which carry some overflow traffic, and to allocate bandwidth to packet channels. Limited trials of the procedure thus far have indicated that the assumptions and simplifications currently incorporated in the method are adequate.

The flow deviation process, by its nature, finds locally optimal solutions. It is necessary to repeat the method with several different random starting designs, leading to different final designs, and then report the best of these.

REFERENCES


