This paper describes an end-to-end performance evaluation model for virtual circuit networks. The model produces mean, standard deviation, and desired percentile of end-to-end response time for each application at each end-user device. The model emphasizes Datakit® II Virtual Circuit Switch-based networks and allows a variety of end-to-end protocols (e.g., SNA/SDLC, 3270 BSC, asynchronous). General, nonsymmetric configurations and traffic patterns are treated.

End-to-end response times are obtained with an iterative solution procedure. The procedure is robust and usually converges in a small number (e.g., <10) of iterations.

The model has been packaged into a performance evaluation tool that is part of the Enhanced Interactive Network Optimization System (E-INOS) DATA and has been used to evaluate the performance of several customers' data networks.

1. INTRODUCTION

Network design and performance analysis tools are needed to design virtual circuit networks to end-to-end application response time constraints. Virtual circuit networks typically consist of a backbone network with point-to-point and/or multipoint access lines, and can support a mixture of synchronous (e.g., SNA1/SDLC, BSC) and asynchronous end-to-end protocols. These protocols can be terminated at the backbone boundary or transported across the backbone. In either case, the backbone is transparent to the end-to-end applications. The virtual circuits can be permanent or switched. As an example, AT&T's CNO 11 offering provides virtual circuit network capabilities using the Datakit® II Virtual Circuit Switch (VCS).

A virtual circuit network (Figure 1) may contain several virtual, multipoint or point-to-point circuits. Each circuit is configured within the backbone network, which is transparent to the end-system hardware. For example, in the SNA case a front-end processor (FEP) controls an SDLC link with one or more cluster controllers (CCs). The FEP executes IBM's Network Control Program (NCP). The SDLC link may be configured as a virtual, multipoint line (VMPL) as follows. The FEP is connected to a backbone switch. The CCs are connected to the backbone by one or more point-to-point or multipoint access lines (these CC access lines are also referred to as tail circuits). All the CCs on all the tail circuits of this single VMPL appear to the FEP as being part of a single physical multipoint circuit. The backbone is transparent to the FEP/NCP and CCs. This transparency is achieved in one of two ways. In the remote polling case, the end-to-end protocol blocks (e.g., SDLC frames) are wrapped in the backbone protocol and transported across the backbone. All polls and acknowledgments traverse the backbone. In the local polling case, the switches that the CC tail circuits are connected to poll the individual CCs. The backbone network transports the user data to the port that the FEP is connected to. The FEP polls this port.

The traffic patterns of the individual VMPLs of a virtual circuit network may be nonsymmetric. The CCs on a
VMPL may have different traffic volumes and mixes of applications. The CC access lines may have different speeds and numbers of drops. The FEP access line may have a higher speed than the CC access lines. Finally, if the end-system protocol units (e.g., SDLC frames) are larger than the backbone protocol units the former may be pipelined into and out of the backbone. With pipelining, a switch port does not wait to accumulate an entire end-system frame before transmitting it into or out of the backbone. This reduces end-to-end delay if these frames are larger than the backbone protocol units.

A model for end-to-end application response time in a virtual circuit network must consider both the end-system and backbone protocols and hardware. A general discussion of models for polled protocols over multipoint private lines is given in [2]. Models for SDLC and 3270 BSC VMPLs are given in [2]-[3], respectively. These models capture many of the details of SDLC and BSC implementation within NCP. The models assume that the line is symmetric. In addition, these models consider a reference connection through a virtual circuit backbone rather than the interactions among several VMPLs and the backbone. The connection between the backbone switch that the FEP is connected to and the switch that each tail circuit is connected to is assumed to be provided by a set of backbone links with known utilizations. It is also assumed that the connection is the same for each tail circuit (i.e., that all the tail circuit connections have the same delays); this is necessary for symmetry. The models allow the FEP access line speed to differ from the tail circuit speeds. The models also assume pipelining into the backbone and allow for pipelining out of the backbone. Various examples using the models are given in [3].

This paper describes a model for end-to-end application response time in a virtual circuit network containing a number of VMPLs with possibly different end-system protocols. The VMPLs are allowed to have nonsymmetric traffic patterns, a variety of tail circuit speeds, and different numbers of CCs on each tail circuit. The model considers interactions among different VMPLs and the backbone network, i.e., backbone performance is analyzed and it is not necessary to specify reference connections and backbone link utilizations. The model considers full and half-duplex SDLC and 3270 BSC VMPLs with remote polling and non-pollled asynchronous connections; however, only the full-duplex SDLC case is discussed here. The backbone network is assumed to be constructed with the Datakit II VCS.

This model has been packaged into a performance evaluation tool that is part of the Enhanced Interactive Network Optimization System (E-INOS) DATA. E-INOS DATA is used by AT&T to design and evaluate the performance of customers' data networks.

The paper is organized as follows. Section 2 contains a brief description of the implementation of SDLC in NCP and the transport of SDLC across a backbone network. Section 3 describes the SDLC performance model. Section 4 describes the iterative procedure for the coupled analysis of the backbone and all the VMPLs. Section 5 contains an example, and conclusions are given in Section 6. The paper does not contain a detailed description of the Datakit II VCS backbone performance models; this work is described in [3].

2. DESCRIPTION OF SDLC IMPLEMENTATION

This section describes the implementation of SDLC within NCP (see [6] for details) and the transport of SDLC frames across a Datakit II VCS backbone. It is assumed the reader is familiar with the SDLC architecture.

In full-duplex SDLC, an ordered list specifies the sequence in which the CCs are serviced. A transmit pointer and a receive pointer cycle through the list. The CC currently pointed to by the receive pointer is polled. The CC sends any information frames it has to the FEP followed by a final frame. The receive pointer is then incremented to the next entry in the list. Likewise, the CC currently pointed to by the transmit pointer is serviced by the FEP. The FEP sends any information frames it has for that CC and increments the transmit pointer. If the FEP is transmitting output to one CC and another CC must be polled, the poll is sent as soon as the current output frame is completed. Transmission of output then resumes. If the FEP is transmitting output to one CC and the same CC must be polled, polling is suspended until output transmission to that CC is completed. If the FEP is polling one CC and output must be transmitted to another CC, the output is sent immediately after the poll. If the FEP is polling one CC and the transmit pointer is incremented to the same CC, the transmit pointer is incremented again (i.e., the CC loses its turn for output).

Several NCP parameters influence the operation of full-duplex SDLC. The parameter PAUSE specifies the minimum duration of the polling cycle, and is used to limit unproductive polling. The MAXOUT parameter specifies the maximum number of output SDLC frames that may be outstanding before an acknowledgment is required. The PASSSLIM parameter specifies the maximum number of output frames the FEP may send to a CC when servicing that CC. Both parameters apply only to output; for input, the maximum number of frames a CC may send when polled depends on the particular CC hardware.

If an error is detected the respective frame must be retransmitted. The effects of an error are to increase the traffic and to cause a delay due to having to wait for retransmission. The model has been extended to include these effects; however, they are neglected here for simplicity.

The transport of end-system protocol units across a backbone network is described in [2]. In the remote polling case, SDLC frames are wrapped in backbone protocol units. The Datakit II VCS uses Universal Receiver Protocol (URP); the protocol unit is the URP block. Supervisory frames are small enough to fit in a single URP block; information frames may require several URP blocks. URP blocks are assembled at the switches where the SDLC access lines connect to the backbone. URP overhead is added, and the blocks are transmitted across the backbone in sequence. Frames comprised of more than one URP block are reassembled at the destination switch.
To improve performance, frames that span several URP blocks are pipelined into the backbone. The URP blocks are assembled as the SDLC frame arrives at the backbone switch from the access line. Each URP block is transmitted into the backbone as soon as it is assembled. In addition, the URP blocks may be pipelined out of the backbone when they reach the destination switch. If this is done it is necessary that an URP block be available for transmission when the last bit of the previous URP block is transmitted otherwise loss of synchronization will result (SDLC is a bit-oriented, synchronous protocol). To avoid this, a 'buildout delay' is applied to the first URP block of the frame. This delay is less than the time required to accumulate all the URP blocks of the frame but sufficiently large to insure with high probability that each URP block is present when needed. The determination of buildout delay is not considered here (see [7]).

The above discussion considers only the SDLC multipoint link, i.e., the SNA route extension network. SNA subarea networks and higher layer effects such as chaining, session pacing, etc. are not considered here.

3. FULL-DUPLEX SDLC MODEL

Consider a VMPL with N CCSs, where Mj applications originate at CC j. Application i at CC j has fixed inquiry and reply message lengths fij and f'ij characters, respectively (for simplicity, only fixed length messages are considered here; the analysis has been extended to the case of gamma-distributed message lengths). Arrivals of inquiry and reply messages are assumed to form Poisson processes with mean arrival rate \( \lambda_i \) for application i at CC j. Each application has host processing time \( H_{ij} \).

End-to-end response time is defined as the elapsed time between the pressing of the enter key after the inquiry message of an application is typed and the appearance of the last character of the reply message on the screen. For application i at CC j, end-to-end response time \( S_{ij} \) is equal to the sum

\[
S_{ij} = w_{ij}^f + t_{ij}^f + H_{ij} + w_{ij}^r + t_{ij}^r,
\]

where \( w_{ij}^f \) and \( w_{ij}^r \) are the input and output-waiting (i.e., queuing) times, respectively, and \( t_{ij}^f \) and \( t_{ij}^r \) are the times to transport the input and output messages, respectively, across the access lines and backbone network. All the quantities in Eq. (3-1) are random variables. The mean and variance of \( S_{ij} \) are of greatest interest. The mean of \( S_{ij} \) is equal to the sum of the means of the quantities on the right hand side of the equation. If correlations among the quantities on the right hand side are neglected, the variance of \( S_{ij} \) is equal to the sum of the variances of these quantities. Throughout this section relations among random variables are used to imply relations among their means and variances; this is done to avoid having to write separate equations for the means and variances. The remainder of this section discusses models for the means and variances of \( w_{ij}^f \) and \( t_{ij}^r \). The mean and variance of \( H_{ij} \) are assumed to be known.

3.1 Input-Waiting Time

Input-waiting time is obtained with a nonsymmetric, exhaustive service, cyclic queueing model [11]. In a cyclic queueing system with \( N \) nodes and exhaustive ser-

vice the server arrives at a queue, begins serving customers, and leaves when the queue is empty. The server then spends a random time, called the walk-time, traveling to the next queue (the queues are ordered). In using a cyclic queueing model to represent polling for input, the walk-time represents the time required for a final frame sent by a CC to traverse the network to the FEP and the FEP to send a poll frame to the next CC, and the service time represents the time for the information frames comprising an inquiry message to traverse the network. The effect of the CC not sending more than some maximum number of frames when polled is neglected.

Let \( b_j^f \), \( \lambda_j \), and \( d_j \) be the service time, mean customer arrival rate, and walk-time for queue j. Let \( C_j^f \) be the input cycle time for queue j, i.e., the duration of the time interval between two successive instants when the server leaves queue j. Given \( \lambda_j \) and the first two moments of \( b_j^f \) and \( d_j \) for each of the queues, the N quantities \( \lambda_j \) are obtained as the duration of the time interval between two successive instants when the server leaves queue j. Given \( \lambda_j \) and the first two moments of \( b_j^f \) and \( d_j \) for each of the queues, the N quantities \( \lambda_j \) are obtained. The mean and variance of \( \lambda_j \), respectively, may be obtained with the algorithm developed in [8] and implemented in [9]. This algorithm requires solving a linear system of size \( N \); previous algorithms required, at best, the solutions of systems of size \( N^2 \). The mean waiting times \( E\{w_j\} \), \( j = 1,...,N \) are obtained from the \( \lambda_j \). Approximations for \( E\{w_j\} \), \( j = 1,...,N \) are obtained by assuming that the cycle times \( C_j^f \) are gamma-distributed [9].

To complete the determination of the input-waiting time moments, the \( \lambda_j \) and moments of \( b_j^f \) and \( d_j \) must be related to the properties of the VMPL. FEP and CC SDLC frame processing delays are neglected to simplify the analysis. Let \( L_{s,f} \) and \( L_{s,i} \) be the FEP access line speed and tail circuit line speed for CC j, respectively; \( \Delta_{p,f} \) and \( \Delta_{p,i} \) be the propagation plus modem-pair delays for the FEP access line and CC j tail circuit, respectively; \( \Delta_{mod,j} \) be the modem turnaround time for CC j; \( D_{f}^p \) and \( D_{i}^p \) be the backbone delays for CC j poll and final frames; and \( \rho_{poll} \) be the length of an RR poll/final frame.

In obtaining the walk-time for CC j, the cases where the FEP is not sending output, is sending an output frame to a CC i \( \neq j \), and is sending output to CC j when it is time to poll CC j must be considered separately. For the first case, the poll is not delayed by output and the walk-time is (the superscript in parentheses indicates the case number)

\[
d_j^{(1)} = \frac{2\rho_{poll}}{L_{s,f}} + \frac{2\rho_{poll}}{L_{s,i}} + 2\Delta_{p,f} + 2\Delta_{p,i} + \Delta_{mod,j} + D_{f}^p + D_{i}^p + \rho_{j},
\]

where \( \rho_{j} \) is the delay the poll experiences at the tail circuit access line for CC j and will be considered shortly. The probability that this case occurs is \( 1 - \rho_{j} \), where \( \rho_{j} \) is the total output traffic utilization (see Section 3.2).

For the second case, the poll frame is delayed by the residual life of the FEP access transmission time of an output traffic frame, averaged over output traffic for all CCSs \( i \neq j \). The walk-time and its probability of occurrence are given by

\[
d_j^{(2)} = d_j^{(1)} + R_{l,j}
\]

(3-3)
and $\sum_{i} p_{r,i}$, respectively, where $R_{t,i}$ is the frame transmission time residual life and $p_{r,i}$ is the output traffic utilization for CC $i$. The first two moments of $R_{t,i}$ are obtained in terms of the first three moments of output SDLC frame size, averaged over all CCs $i \neq j$.

For the third case, the poll frame is delayed by the residual time to finish serving CC $j$ for output. The walk-time and its probability of occurrence are given by

$$d_f^{(3)} = d_f^{(1)} + R_{a,j}$$

(3-4)

and $p_{r,j}$, respectively, where $R_{a,j}$ is the residual CC output service time (see Section 3.2).

The delay $d_f^{(3)}$ occurs because the FEP can send output frames to other CCs on the tail circuit while CC $j$ is being polled [2]. This delay depends on whether the FEP access line speed is equal to or greater than the tail circuit speed. In the former case a queue will not develop; the poll will have to wait for the buildup plus transmission on the tail circuit of one URP block. In the latter case a queue can develop; the delay is the M/G/1 waiting time due to output traffic sent to CCs $i \neq j$ on the tail circuit.

The arrival rate for CC $j$, $\lambda_{j}$, is the sum of the arrival rate for the individual applications. The input message service time is the time to transmit a message on the CC access line averaged over all applications at CC $j$. This is given by

$$b_{j} = \frac{\sum_{i=1}^{M} \lambda_{i} f_{j} + \left[ f_{j}/p_{f} \right] f^{OH}}{\lambda_{j} - \beta_{j}}$$

(3-5)

where $f_{j}$ and $f^{OH}$ are the maximum size of the data portion of an inbound information frame and the amount of SDLC overhead per frame, respectively. Eq. (3-5) assumes that the tail circuit speed is less than or equal to the FEP access line speed. The backbone and propagation delays are not included in input service time; these are accounted for in the walk-time because the information and final frames are pipelined, i.e., are sent in succession and are in various stages of transit simultaneously.

The effect of the PAUSE parameter may be accounted for by modifying the mean and variance of $d_{1}$, the walk-time between CCs $N$ and 1. This analysis will not be discussed here.

3.2 Input and Output Transport Time

Input message transport time is equal to the sum of the message transmission time on the tail circuit, the propagation plus modem-pair delays on the tail circuit and FEP access line, the backbone delay, and the transmission time on the FEP access line for the smaller of the entire message and a single URP block. The result is

$$t^{(j)}_{f} = \frac{\left( f^{(j)}_{i} + \left[ f^{(j)}_{i}/p_{f} \right] f^{OH} \right)}{L_{s,j}} + \Delta_{p,j} + \Delta_{p,t}$$

$$+ D^{(j)}_{f} + \min \left( \frac{L_{URP}}{L_{s,t}} \right)$$

(3-6)

where $D^{(j)}_{f}$ is the input message backbone delay for application $i$ at CC $j$. Output message transport time is obtained in a similar manner, with the superscript $e$ replaced by $r$ to refer to output and with an additional term $g^{(j)}_{f}$ to represent the delay of the last URP block of the message at the switch the tail circuit is connected to. If the FEP access line speed is equal to the tail circuit speed, this delay is equal to the poll delay $g^{(j)}_{f}$. If the FEP access line speed is greater than the tail circuit speed, this delay is equal to the M/G/1 waiting time at the tail circuit due to output traffic sent to all CCs on the tail circuit.

3.3 Output-Waiting Time

Output-waiting time is obtained with a zero walk-time, nonsymmetric, exhaustive service, cyclic queueing model. It is shown in [9] that the quantities $X^{(j)}_{i}$, $j = 1, ..., N$ (see Section 3.1) remain finite as the walk-times approach zero. The output message service time for CC $j$, $b^{(j)}_{f}$ is equal to the FEP access transmission time averaged over all the applications at CC $j$. The output traffic utilization is $\lambda_{j} b^{(j)}_{f}$.

The interspersal of poll frames between output frames may be treated with a vacation model. This has been done by assuming that the time between polls is exponentially distributed and replacing message service time by message completion time in the cyclic queueing model [10]. The effect is usually small and the analysis is fairly lengthy; therefore, it is not discussed here.

The input-waiting time model requires residual CC output service time. Moments of this quantity are given by

$$E[R^{(j)}_{a,j}] = E[R^{(j)}_{a,j}] + \left( n + 1 \right) E[R_{j}]$$

(3-7)

where $R_{j}$ is the total time to serve CC $j$. The moments of $R_{a,j}$ are averaged over all cycles. In the zero walk-time case these moments are equal to zero because there are an infinite number of cycles of zero duration. However, the moments of $R_{a,j}$ remain determinate as walk-time and cycle time approach zero. The first two moments of $R_{j}$ are related to cycle time moments by [11]

$$V[R_{j}] = \rho_{j} \cdot E[C^{(j)}_{j}]$$

(3-8)

$$\text{Var}[R_{j}] = \rho_{j}^{2} \text{Var}[C^{(j)}_{j}]$$

where $C^{(j)}_{j}$ is the output cycle time. Substituting Eqs. (3-7)-(3-8) into the expression for $E[R_{a,j}]$ and letting the moments of $C^{(j)}_{j}$ approach zero produces

$$E[R^{(j)}_{a,j}] = \frac{1}{2} \rho_{j} \cdot \frac{E[b^{(j)}_{f}]^{2}(1-\rho_{j}^{2})}{2E[b^{(j)}_{f}]^{2}(1-\rho_{j}^{2})^{2}}$$

(3-9)

The second moment of residual output service time requires the third moment of $R_{j}$, which is obtained by assuming $R_{j}$ is gamma-distributed. The result is

$$E[R^{(j)}_{a,j}] = 8E^{2}[R_{a,j}] / 3$$

(3-10)

4. ITERATIVE SOLUTION FOR END-TO-END PERFORMANCE

The models for input and output waiting time are coupled through the backbone and the residual CC output service times. The models are nonlinear and it is not possible to obtain closed-form expressions for waiting time moments for all CCs of all VMPLs. Instead, the iterative solution procedure shown in Figure 2 is used. The first step is
INITIALIZATION

FOR ALL VMPLs {
    FOR ALL CCs {
        ARRIVAL RATES
        INPUT AND OUTPUT SERVICE TIME MOMENTS
    }
    POLLING RATE = 0
}

A

BACKBONE ANALYSIS [5]

TRAFFIC
DELAYS

DO THROUGH B FOR ALL VMPLs

VMPL ANALYSIS

FOR ALL CCs {
    OUTPUT-WAITING TIME MOMENTS
    WALK-TIME MOMENTS
    INPUT-WAITING TIME MOMENTS
}

B

DOES MAXIMUM WAITING TIME MOMENT FRACTIONAL CHANGE EXCEED THRESHOLD?

NO

YES

STOP

Figure 2. Iterative Solution Procedure.

initialization. Application arrival rates and first two moments of input and output service time are obtained for each CC of each VMPL. In the second step, backbone analysis is performed [5]. A backbone traffic stream is created for the inquiry and reply traffic associated with each CC and the poll and final traffic associated with each tail circuit. The first two moments of backbone delay are obtained for each traffic stream (backbone routing must be known). The third step is to analyze each VMPL in succession. The first two moments of output waiting time are obtained. Next, for each CC the moments of delay a poll experiences at the respective tail circuit are calculated. After this, Eqs. (3-2)-(3-4) are used to obtain input walk-time moments. Then, input-waiting time moments are obtained. In the fourth step, the abso-
lute fractional change relative to the previous iteration is calculated for all waiting time moments. If the largest fractional change exceeds a preset threshold, another iteration is performed beginning with the second step. If this change is less than the threshold, the iterative procedure terminates. Input and output transport times are obtained after termination.

The stability of the iteration has not been established for all possible networks as the models are highly nonlinear and the number of unknowns can be large. However, the iteration has been found to converge in all cases analyzed to date. These cases involve mixtures of various end-to-end protocols and Dat Kit backbone network configurations. They include both symmetric and nonsymmetric VMPLs. Also, convergence has been obtained both for cases with heavy background traffic (specified via a traffic matrix) and no background traffic with either heavy or light loads specified at each cluster controller. Convergence for the latter case indicates that the model is robust because, with light loads and zero background traffic, most of the traffic is due to poll and final frames and the backbone and multipoint circuit models are highly coupled.

5. EXAMPLE

An example network is shown in Figure 3. The four tail circuits with 15 CCs are on a single VMPL controlled by the FEP. There are four applications, which are summarized in Table 1 (the applications have zero host processing time). Traffic volumes are 0.0278 s⁻¹ (100 hr⁻¹) for CCs 1-4 and 12-15, 0.0139 s⁻¹ (50 hr⁻¹) for CCs 5-9, and 0.0208 s⁻¹ (75 hr⁻¹) for CCs 10-11. Propagation plus modem-pair delays are 0.03 s for the tail circuits and zero for the FEP access line. Modern turnaround times are

![Figure 3. Network for Example of Section 5.](image-url)
6. The performance of VCS-based backbone networks that transport a variety of protocols, configurations and traffic patterns. The effect of asymmetry is directly observable in the standard deviation by assuming response time is gamma-distributed. The overall model handles general, nonsymmetric applications and from the technical point of view, the two backbone solutions are compared. The Datakit II VCS backbone performance models used in the overall model have been used to evaluate several customers' networks.

7. ACKNOWLEDGMENTS

The author would like to acknowledge S.-C. Lin for developing and implementing the Datakit II VCS backbone performance models used in the overall model described here. The author would also like to thank R. Cole, E. Hernandez, and R. Moats for providing detailed information on the Datakit II VCS; R. Cole and J. Medama for useful discussions on modeling the SDLC implementation in NCP; W. Strain and J. Whitehead for useful feedback on model features and results; and D. Sarkar for discussions on cyclic queues.

FOOTNOTES

1. SNA is a trademark of the IBM Corporation.
2. The Datakit II VCS is a product of AT&T.

REFERENCES


Table 1: Applications for Example of Section 5.

<table>
<thead>
<tr>
<th>APPLICATION</th>
<th>MESSAGE LENGTH (char)</th>
<th>INPUT</th>
<th>OUTPUT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100</td>
<td>400</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>100</td>
<td>1000</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>50</td>
<td>2000</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1000</td>
<td>1000</td>
<td></td>
</tr>
</tbody>
</table>

0.15 s for the 9.6 kbps tail circuits and 0.05 s for the 4.8 kbps tail circuit. The two backbone links are loaded with background traffic. This traffic consists of application 4 with a volume of 139 s⁻¹ (5×10⁸ hr⁻¹).

Results are shown in Table 2. Convergence with a threshold of 1.0×10⁻⁷ required five iterations. Ninety-fifth percentile of response time was obtained from mean and standard deviation by assuming response time is gamma-distributed. The effect of asymmetry is illustrated by the variation in the mean response time for application 2. This response time is considerably higher at CC 7 due to the lower tail circuit speed. In addition, the mean response time for application 3 is slightly higher at CCs 10 and 11 than at CCs 12-15 because traffic to and from the former must traverse a backbone link while traffic to and from the latter is routed by a single backbone node directly to the appropriate tail circuit. The effect is reduced due to the smaller traffic load of the tail circuit for CCs 10 and 11 compared to the delay of CCs 12-15. Finally, the results show that there is very little variation in the standard deviation of response time for this example.

Table 2: Results for Example of Section 5.

<table>
<thead>
<tr>
<th>CC</th>
<th>APPLICATION</th>
<th>MEAN (s)</th>
<th>STANDARD DEVIATION (s)</th>
<th>95th PERCENTILE (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>3.38</td>
<td>1.46</td>
<td>6.09</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>3.39</td>
<td>1.47</td>
<td>6.10</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>3.39</td>
<td>1.47</td>
<td>6.11</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>3.40</td>
<td>1.47</td>
<td>6.12</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>3.40</td>
<td>1.48</td>
<td>6.14</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>3.94</td>
<td>1.48</td>
<td>6.64</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>5.11</td>
<td>1.47</td>
<td>7.75</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
<td>3.93</td>
<td>1.49</td>
<td>6.65</td>
</tr>
<tr>
<td>9</td>
<td>2</td>
<td>3.93</td>
<td>1.49</td>
<td>6.65</td>
</tr>
<tr>
<td>10</td>
<td>3</td>
<td>4.81</td>
<td>1.49</td>
<td>7.49</td>
</tr>
<tr>
<td>11</td>
<td>3</td>
<td>4.81</td>
<td>1.48</td>
<td>7.48</td>
</tr>
<tr>
<td>12</td>
<td>3</td>
<td>4.77</td>
<td>1.48</td>
<td>7.44</td>
</tr>
<tr>
<td>13</td>
<td>3</td>
<td>4.77</td>
<td>1.48</td>
<td>7.43</td>
</tr>
<tr>
<td>14</td>
<td>3</td>
<td>4.76</td>
<td>1.47</td>
<td>7.41</td>
</tr>
<tr>
<td>15</td>
<td>3</td>
<td>4.75</td>
<td>1.47</td>
<td>7.39</td>
</tr>
</tbody>
</table>

6. CONCLUSION

The model described in this paper allows end-to-end performance evaluation on an application basis for Datakit II VCS-based backbone networks that transport a variety of protocols. The model handles general, nonsymmetric configurations and traffic patterns. The solution procedure is robust and converges in a small number of iterations. The model has been used to evaluate several customers' networks.