Eliminating the Last Mile Bottleneck

P. Michael Henderson
michael.henderson@cox.net

William Lipp
Lipp Service Consulting[1]
w.b.lipp@ieee.org

August 30, 2000

Bandwidth in the network backbone, both packet switched and circuit switched, is growing by leaps and bounds – reliable experts estimate that total backbone bandwidth doubles every nine months. Unfortunately, bandwidth to the customer is not growing at this same rate. Replacing the local copper loop with fiber is expensive and will occur over a long time period, probably decades. Until this occurs, the challenge for network designers will be to maximize the capacity of the existing local copper loop.

In one dimension, this is already in progress. Recent developments in digital subscriber line (DSL) have increased the achievable data rates over copper. The new HDSL2 standard, for example, will allow T1 or E1 operation over a single twisted pair, something that took two or three twisted pairs with earlier technology.

Although some additional gains may be achieved with better modulation techniques, or by finding ways to cancel crosstalk from other high-speed services, significant additional gains are probably not possible. Other avenues must be explored to achieve our goals of increasing local loop capacity.

Techniques which exploit signal processing are especially attractive because of the declining cost of silicon. The cost of processing is a one-time capital cost to the provider as compared to the recurring charge for an additional twisted pair from the telephone company. Techniques which increase local loop capacity, such as the one we describe here, may prove especially attractive to competitive local exchange carriers (CLECs) as they seek to offer bundled services to business and residential customers.

One promising technique is to mix voice and data over a DSL line, especially an HDSL line used for business access. A low delay speech coder will allow many more voice lines to be carried over the DSL line. Significant additional capacity can be attained by utilization of silence suppression on the voice lines so that bandwidth is utilized only when speech is active.

The characteristics of voice and data complement each other well in the access environment. Voice lines are sparsely utilized with predictable peaks of usage during the day. But when a voice line is in use, the voice information must be communicated with low delay since delay is one of the major factors in perceived voice quality. Data access, on the other hand, can generally tolerate delay. Even non-interactive streaming media can be buffered for the worst case delay.

The challenge for the system designer is to specify a system which will meet the performance requirements under the worst case anticipated load. This paper proposes a set of performance requirements and discusses how the analysis can be done.

[1] Whenever William Lipp provides consulting services, the client pays Lipp Service. Here, William performed the non-closed form simulations described in this report.

Performance Requirements

Voice over DSL (VoDSL) can be used for both business and residential services. The business environment is likely to utilize many voice lines while the residential environment may be limited in the number of voice lines supported, probably limited to less than ten. Because of this, we focused our initial analysis on the business environment. A future paper may analyze the residential environment.

The primary performance requirement for the system is the worst-case delay for voice. People are extraordinarily sensitive to delay in speech. Complaints are heard about conversations over cellular phones which introduce significantly less than 100 milliseconds of delay, one way. Since conversations over a VoDSL system may include a cellular leg, the additional delay attributed to the VoDSL system must be kept to a minimum.

To determine the acceptable delay over the DSL link, one of the authors (Henderson) contacted companies involved in VoDSL. They indicated that their customers, the service providers, were imposing a limit of 20 ms, one way, on the VoDSL companies. This was established as the target for our system design[2].

Within the system, there are many sources of delay. The primary elements are the coder delay, queueing delay, transmission delay, and decoder delay. Coder and decoder delay are determined by the choice of coder, and will be discussed later. Transmission delay is determined primarily by the type of DSL. In the business environment this is likely to be HDSL, with a transmission delay of less than 1 ms. The value selected for the queueing delay must be such that the one-way delay stays within the 20 ms target set previously.

The maximum possible value of queueing delay depends upon the coder selected, which has not been discussed yet. Too small a value will limit the number of simultaneous conversations, while too high a value could cause us to exceed our limit of 20 ms one way delay. Based on these constraints, and in order to provide some delay margin, we chose a queueing delay of less than 10 ms, 99% of the time, under the worst-case load.

The caveat, “99% of the time,” is extremely important. Since we’re operating with statistics, the tail of the distribution could extend out quite far. If the requirement was that the queueing delay could never exceed 10 ms, the number of voice lines supported would be quite small. Setting the bar at 99% of the time means that we have to develop techniques for the 1% of speech frames which are delayed more than 10 ms in the queue (under worst case load).

The problem of frame erasure concealment has been addressed in the literature and solutions are available. Newer techniques are presently under development.

Looking at the data side of the link, it is important that bandwidth is always available for Internet access. Establishing a minimum, or guaranteed, data rate for data will minimize the number of complaints from computer users.

[2] Although we do not address echo cancellation in this paper, it is important to note that 40 ms of additional round trip delay may be objectionable to certain subscribers. In general, local exchange companies do not insert echo cancellers in local calls because the delay is generally below the perception threshold. With this additional delay, however, this may not be true. Echo cancellers should be included in real world VoDSL systems.

Finally, voice lines provided to a business are subject to being used for fax machines or data modems. Not only must these uses be supported, but the impact of their use on the maximum number of lines must be analyzed.

Summarizing then, the performance requirements established are:

1. The maximum one-way speech delay over the DSL portion of the communications link must not exceed 20 ms.
2. The maximum queueing delay, under worst-case load, must not exceed 10 ms, 99% of the time.
3. It must be possible to reserve bandwidth for data.
4. The voice circuits must support fax and modem usage, although their usage may reduce the maximum number of voice circuits supported.

System Architecture

The system architecture which we analyzed is shown in figure 1. An HDSL line is shown carrying ATM cells. A number of speech coders are utilized, one for each voice circuit, on each end of the ATM circuit. Voice frames utilize ATM adaptation layer 2 (AAL2) while Internet data utilizes AAL5. The delay through the ATM network is not modeled in this analysis (the ATM network delay is assumed to be zero). For real world applications, the worst-case delay thorough the ATM network should be added to the delay over the DSL line.

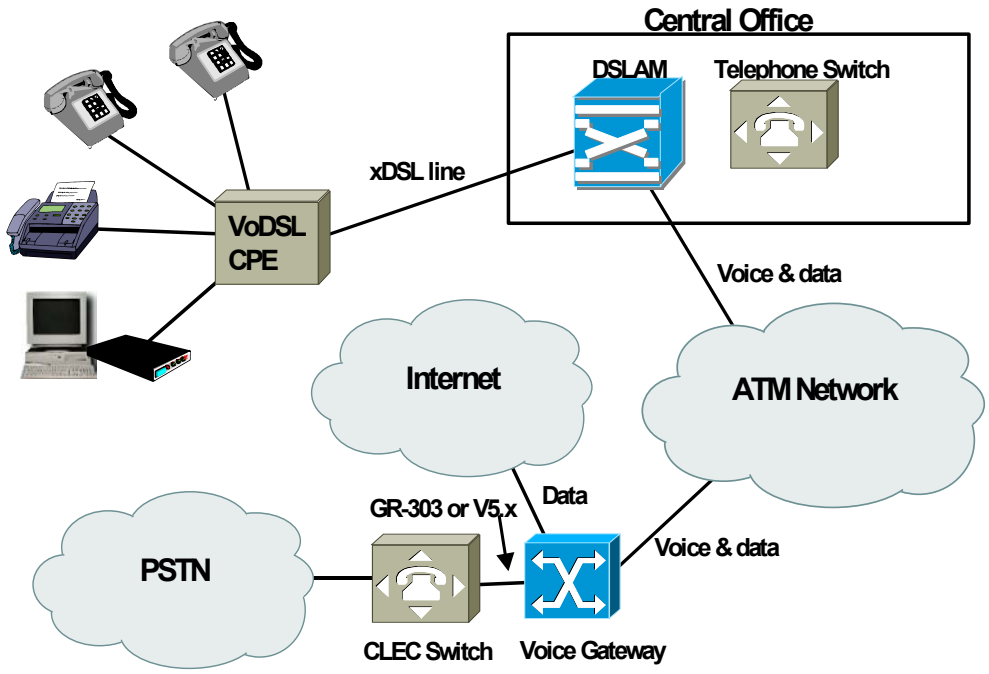

Figure 1: Network architecture for provisioning VoDSL. Note that the ATM cells simply flow through the DSLAM and are carried to the gateway over the ATM network.

Figure 2 illustrates the system from a queueing analysis point of view. N voice coders output their speech frames to the AAL2 processor which then queues the cells for transport by the line. For the closed form analysis, the AAL2 processor is assumed to be infinitely fast and can process voice frames as fast as they arrive – queueing does not occur at the AAL2 processor (this simplifying assumption is removed for the non-closed form analysis). The line appears as a single server queueing system, therefore, with two queues, one for AAL2 voice frames and one for AAL5 Internet traffic. The server utilizes an algorithm which allows it to provide a guaranteed minimum amount of bandwidth to the Internet traffic. For our analysis, we assume that the AAL5 queue is always full so that the AAL2 cells may not utilize any of the guaranteed data bandwidth. This is a worst case assumption but winds up making the analysis simpler – we can essentially remove the bandwidth allocated to data, and the data, when analyzing the voice portion of the system.

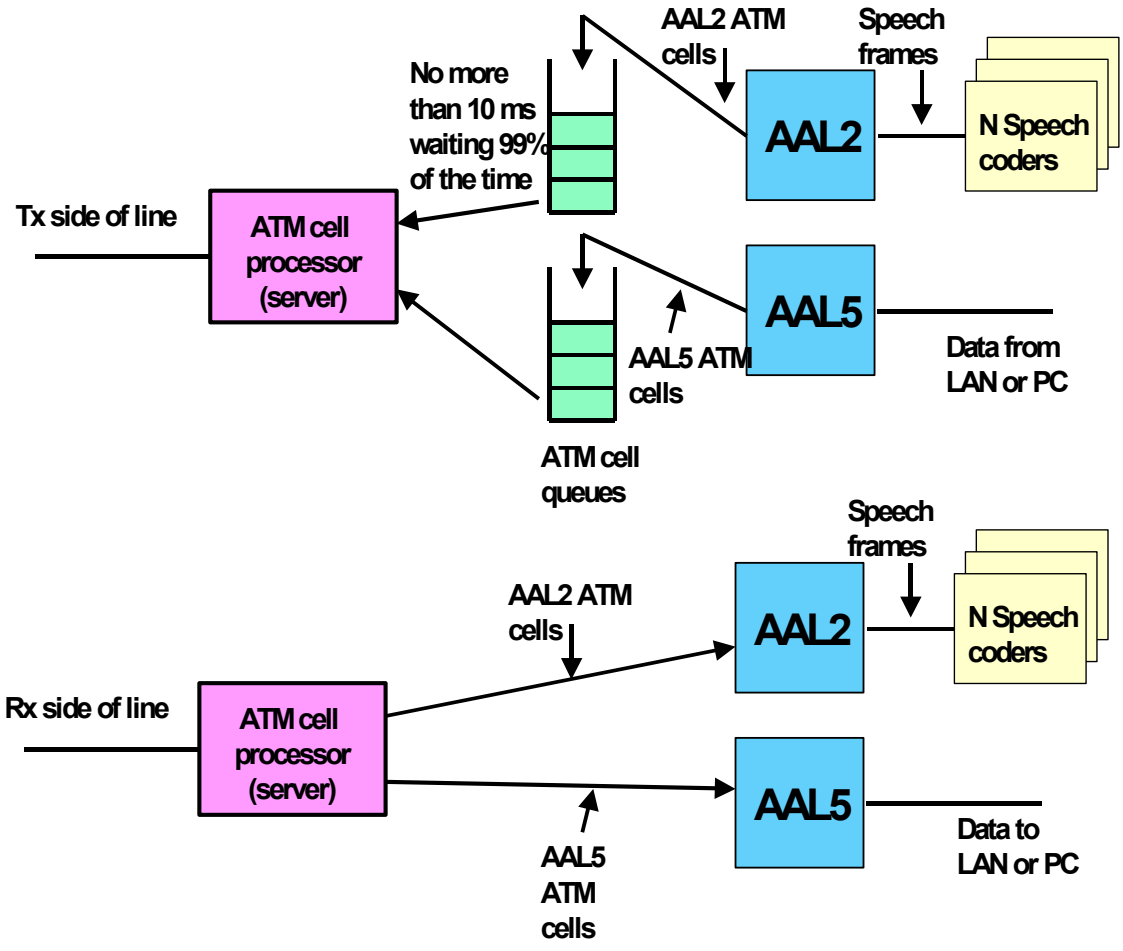

Figure 2: VoDSL queueing system to be analyzed.

For each configuration, we want to solve for "N," the number of voice coders which can be supported, subject to the performance requirement conditions established earlier.

The Speech Model

Everyone knows intuitively that speech is not continuous – it simply is not possible for two people to talk simultaneously and communicate effectively.

But what are the real characteristics of speech? On average, how much time is spent talking and how much time listening? Even when we speak, we do not speak constantly – we pause between phrases and sometimes even between words. And different people speak differently. How, then, are we to model speech?

This question has been studied for many years because of the telephone network. Brady's model [BRA69] has been used for years and shows that talk spurts are exponentially distributed, with talk spurts averaging 352 ms and silence periods averaging 650 ms. Deng [DEN95], however, published a study which showed that talk spurts were much longer, averaging over 7 seconds for conversation. Closer examination indicated, however, that Deng had used a personal computer microphone for his tests, which was probably sensitive to background noise.

Research in various Speech Labs have indicated that the model is very dependent on the voice activity detection (VAD) algorithm in the speech coder, which varies from implementation to implementation, but is much closer to Brady's model than Deng's. Based on this research, we decided to use the classical model for our analysis.

The Speech Coder

The speech coders considered for this system were G.711, G.726, G.728, and G.729. G.723 was not considered because its 30 ms frame size exceeded our limits for delay across the DSL link, even before any other delays were considered. G.729 was evaluated as marginal because of its 10 ms frame[3] but we decided to evaluate it for completeness.

Closed Form Solution

We decided to analyze this system using the standard closed form M/M/1 model to obtain a first approximation of the performance of the system. The M/M/1 model assumes independent Poisson arrivals and exponential service times. Compared to the average talkspurt, however, the speech frames are small. Arrivals are not independent, therefore. Once an utterance begins, the probability is that it will be continued at the time scale we're working at.

The closed form has the advantage of ease of solution. For this work, we used Excel spreadsheets available with [GRO98], modified to our needs. Frame sizes were established as follows: for G.711 and G.726 (32Kbps), 20 octet frames; for G.728 and G.729, 10 octet frames. The larger frame sizes were used for G.711 and G.726 to minimize overhead. Smaller frame sizes were used with G.728 and G.729 to minimize delay.

[3] In coding a frame, the speech coder must generally wait one frame time for the utterance to be buffered. Then it must process the frame, and the frame must be decoded at the receiving end. This can be as much as 3 frame times but is usually 2 to 2.5 frame times. G.729 could introduce a coder/decoder delay of 20+ ms, therefore, more than our link delay budget.

The classical speech activity model indicates that speech is active about 35%, on average (352 divided by (352+650)) so we started with 35% as the voice activity. Since the arrivals of the speech samples are not independent, we expected this level of voice activity to overstate the actual capacity of the line.

Since this analysis was exploratory to determine the effects of different speech coders, the only line rate analyzed was 768Kbps.

G.711

For G.711, we collect 20 octets per line before sending the voice frame to the AAL2 protocol processor. AAL2 appends three octets per voice frame. Taking other AAL2 overhead into consideration, this allows a bit more than 2 frames per cell. Under light loading, it is possible that the second voice sample might not arrive for some relatively long period of time. Therefore, the system was set up to hold the cell no longer than 5 ms, after which it will be sent half empty.

The time required to buffer 20 octets of G.711 is 2.5 ms (silence suppression is done on these 2.5 ms boundaries so that the system only deals with fixed length blocks). The end to end delay, therefore, is 2.5 ms to buffer the voice, 10 ms maximum queueing delay, and 1 ms transmission time, for a total of 13.5 ms. Decode time is assumed to be zero. The 5 ms buffering delay is not a factor because under heavy load, the cells are never held – they are sent as quickly as possible. And under light load, there is no queue time.

Since the system is designed to limit the length of the queue (less than 10 ms delay 99% of the time), the server is not busy 100% of the time handling voice. The voice non-busy time can be used for data transmission, increasing the average bandwidth available to data. Note that this bandwidth is an average – at any instant of time, the only bandwidth that is guaranteed to data is that indicated as "guaranteed." However, on time scales which are realistic for data, during the busy hour, the average data rate is available. See table 1 for the results for G.711.

Guaranteed data rate (Kbps)	Number of voice ports (G.711)	Average queueing system delay (ms)	99% system delay (ms)	Average data rate (Kbps)
64	18	2.34	8.73	245.1
128	16	2.42	8.94	303.2
192	14	2.50	9.20	361.3
256	12	2.59	8.70	419.4
384	8	2.80	9.39	535.6
512	4	3.03	7.45	651.8

Table 1: Number of ports supported with G.711 coding, silence suppression, and 35% speech activity, on a 768Kbps HDSL line. Packetizing delay (to accumulate 20 octets) is 2.5 ms for G.711.

G.726 (32 Kbps)

G.726 is treated the same as G.711. The only difference is that it takes 5 ms to accumulate 20 octets. The worst case delay, therefore, is 5 ms to collect the 20 octets, 10 ms for the worst case queueing delay, and 1 ms for transmission, for a total of 16 ms. The results are given in table 2.

Guaranteed data rate (Kbps)	Number of voice ports (G.726)	Average queueing system delay (ms)	99% system delay (ms)	Average data rate (Kbps)
64	37	2.54	9.94	230.6
128	33	2.64	9.61	288.7
192	29	2.74	9.94	346.8
256	24	2.59	8.70	419.4
384	16	2.80	9.39	535.6
512	9	3.38	9.11	637.3

Table 2: Number of ports supported with G.726 coding, silence suppression, and 35% speech activity, on a 768Kbps HDSL line. Packetizing delay (to accumulate 20 octets) is 5 ms for G.726.

G.728

The problem with low bit rate coders is that too much delay is introduced by buffering 20 octets. G.728 requires 10 ms to produce 20 octets, making this approach undesirable.

In this situation, we have chosen to use 10 octets, or 5 ms of speech, for the G.728 coder. As before, AAL2 adds three octets of overhead per voice frame. This allows each cell to carry speech samples for a bit less than four voice circuits. We don't want to hold a cell indefinitely waiting for four speech samples because this would add too much delay. We have chosen to hold the cell for a maximum of 5 ms. If four speech samples are not available in that time, the cell is sent partially empty. Again, the analysis is done so that 99% of the time, the queueing delay is less than 10 ms, under the heaviest load. The worst-case delay is the sum of the 5 ms to collect 10 octets, the 10 ms 99% queueing delay, 1 ms for transmission, and 1 ms to begin the decoding, for a total of 17 ms one way. The 5 ms worst case waiting delay is not additive because under worst case loading, there will always be a queue and the next cell will be sent as soon as the previous one is transmitted (there is no need to hold cells).

The result of the analysis is indicated in table 3, below.

Guaranteed data rate (Kbps)	Number of voice ports (G.728)	Average queueing system delay (ms)	99% system delay (ms)	Average data rate (Kbps)
64	66	2.61	9.94	226.2
128	58	2.59	9.61	291.9
192	51	2.70	9.94	349.3
256	43	2.67	9.52	415.0
384	29	2.91	9.39	529.9
512	16	3.40	9.11	636.7

Table 3: Number of ports supported with G.728 coding, silence detection, and 35% speech activity, on a 768Kbps HDSL line.

G.729

G.729 is analyzed in the same manner as G.728. For G.729, 10 octets are accumulated every 10 ms. However, G.729 operates differently than G.728. G.728 operates on speech in 625 microsecond "chunks", outputting a frame in 2.5 ms. G.729 actually operates on 10 ms "chunks" so it must wait until 10 ms of speech has accumulated before processing it. The end-to-end delay, therefore, is about 15 ms to accumulate 10 octets, 10 ms for 99% queueing delay, 1 ms for transmission, and about 5 ms for decoding, for a total of about 31 ms, one way. This extra delay pays off, however, in a larger number of lines supported, as can be seen in table 4, below.

Guaranteed data rate (Kbps)	Number of voice ports (G.729)	Average queueing system delay (ms)	99% system delay (ms)	Average data rate (Kbps)
64	132	2.61	9.94	226.2
128	116	2.59	9.61	291.9
192	103	2.77	9.84	345.2
256	87	2.74	9.52	410.9
384	59	2.99	9.39	525.8
512	32	3.40	9.11	636.7

Table 4: Number of ports supported with G.729 coding, silence detection, and 35% speech activity, on a 768Kbps HDSL line.

For reference, the estimated one-way delay for each of the speech coders is given in the table below.

Speech coder	Maximum number of voice lines	Worst case delay (ms)
G.711	18	13.5
G.726 (32Kbps)	37	16
G.728	66	17
G.729	132	31

Table 5: Worst case one-way delay for the indicated speech coders and the system design described in the text. Max lines are also shown to indicate the trade-off between number of lines supported and delay.

Non-closed Form Solution

Based on the analysis done with the closed form solution, we decided to further analyze the system with simulation, using MathCAD. We chose to limit our analysis to the G.728 coder because of its relatively low data rate, high quality, and acceptable delay characteristics.

The simulation was done in two phases. In the first phase we assumed only voice traffic on the voice circuits. In the second phase, we analyzed the effect of some of the voice circuits being used for fax. The fax analysis and results are discussed in a latter section.

The performance of queueing systems is strongly affected by the variations in the processes. For example, a classic Queuing Theory result, the Pollaczek-Khinchin formula, shows that the average delay depends on the variance of the service time. There are several important ways that the variation of the real system differs from the variation in the M/M/1 queueing model:

1. ***Persistence of High Loads***. Consider chopping time up into 5ms segments, and examining the number of arrivals within successive segments. For the M/M/1 model, if there are many arrivals in one segment, that doesn't increase the expected of number of arrivals in the next segment. Although the queue increased during the first segment, The M/M/1 queue is likely to decrease during the next segment. But in the real system, a high number of arrivals means many people are talking. Those same people are very likely to be talking during the next segment, so the queue is likely to grow even larger.

2. ***Short Term Counterbalance for Peak Arrivals***. In the real system, if there are 40 voice ports and 40 speech samples arrive within a 3ms period, zero speech samples will arrive in the next 2 ms period, because every speaker is encoded once in a 5ms period. In the M/M/1 system, the 2 ms period would be just another, typical 2 ms period.

3. ***Long Term Limitations on Peak Arrivals***. Over any time period, there is an absolute limit to how many voice samples can arrive in the real system: one every 5 ms for every voice port.

4. ***Service Time Variability***. The number of speech samples the real system can carry in a time period is essentially constant. The variation comes from edge effects - whether the scheduler allocated the last cell to voice or data, and whether the last voice sample fit within a cell or was split between cells. But these edge effects do not accumulate; combining time periods will erase the effects at interior edges, leaving only the remaining edge effects. In the M/M/1 queueing system, the variance within periods adds when you combine those periods.

Two mathematical models were combined in order to approximate the variation of the real system while keeping the analysis tractable: a Block-Matrix Markov Queueing model and a Jitter model.

To track the persistence of high loads, the queueing model uses a state variable for the number of active speakers. To mimic the long-term service time variability, the queueing model uses constant service times. To keep the analysis tractable, the queueing model uses a Markov chain embedded at the completion of service opportunities (if there is no speech sample to transmit, the service is unused, but another opportunity is not available until one service time later).

Given these choices, a model is needed for the number of arrivals within a service time given the number of active speakers. Exactly matching the long-term variability would require one or more additional state variables to mimic the short-term counterbalance effect described above. These additional state variables would have increased the complexity of the modeling and analysis. A simpler model was chosen, with "n" or "n+1" arrivals during a service time. The value of "n" and the probabilities of "n" versus "n+1" were chosen to match the average arrival rate given the number of active speakers. This models the persistence of high loads and keeps the long-term variance of loads reasonably low.

It makes sense to exclude the edge effects of short-term variability from the queueing model because these variations do not accumulate, but the effects are nevertheless real, and should be modeled somewhere. The queueing model is used to determine the time between the arrival of a speech sample and the beginning of transmission for that speech sample. An independent Jitter Model is then used to model the short-term edge effects. There are three major effects:

1. A speech sample inserted into the early part of an ATM cell is not transmitted until the entire cell is ready.
2. A speech sample that is inserted into the end of a cell may be split, and will not be sent until the following AAL2 cell is ready.
3. A speech sample that is split may have one of more data cells transmitted before the next AAL2 cell is transmitted.

Queueing Calculations

These modeling choices lead to a simple block-matrix structure for the Markov chain. Each block is a square matrix, with rows and columns representing the number of active voices. A system with sixty voice ports has matrix blocks that are 61x61, representing the states of zero to 60 active voices. The entire state transition matrix has a simple pattern of repetitive matrices on the diagonals. These structures are known as Quasi Birth-Death (QBD) processes, solution techniques for which are described in [NEU95]. To apply the QBD methods to larger systems, the blocks must be grouped together, so the 61x61 matrices are grouped as 122x122 matrices. These details are also covered in [NEU95]. The result of these calculations is the joint probability distribution for the number in the queue and the number of active voices at each embedding point.

Jitter Calculations

The jitter model is used to capture the effect of the short-term variations that are ignored in the queueing calculations. The queueing model gives the delay until the start of the "service" for a voice sample. This start of service can be anywhere within the 47 octets that are the AAL2 payload. If the first octet of the speech sample is between octet 1 and octet 35 of the AAL2 payload, the jitter delay is the time until the end of that cell. But if the speech sample (with the 3 octet header) is inserted more than 35 octets into the 47 octet cell payload, it will be continued to the next cell and the jitter delay will be the time until the cell containing the rest of the speech sample is transmitted. This introduces the possibility of data or fax cells being transmitted between the beginning and the end of the voice sample.

When there is only data and voice, simple schedulers will insert "n" or "n+1" data cells between voice cells, such that the correct average rates are maintained. When there is voice, data, and fax, then the possibilities are more complex and will be discussed in the fax section.

One remaining component of the delay time is the actions between the embedding points for the queueing model. When there are multiple arrivals in an interval, the later voice samples have additional delay waiting for the earlier voice samples. And all voice samples see a delay between their actual arrival time and the next embedding instant in the model. These delays are added to the other delays.

The probabilities from the queueing model are at embedding points – the idealized service times. Multiple arrivals can happen between embedding points. Later arrivals must wait for the service of earlier arrivals, as well as the service of all voice samples that were in queue at the previous embedding point. The queueing model calculates joint probabilities of queue size and active voices, so this effect is straightforward to calculate.

Total Delay

There are four components of delay that must be added together:

1. Queueing delay for arrivals from prior periods.

2. Queueing delay for prior arrivals in the same period.
3. Jitter.
4. Delay from arrival until next embedding instant.

Components 1-3 are independent. Component 4 has a worst-case value of 1 period. These delays are added together to determine the probability of total delay exceeding 10 ms.

AAL2 Padding

If the AAL2 packing process has a partially filled ATM cell and no more voice samples, it will wait up to 5ms for another voice sample. If no voice samples are received in that time, the cell is padded and sent. When there are four or more active voices, padding is never triggered. For one active voice, the system gets into a 10 ms cycle that sends 21 octets of padding every cycle:

T=0 ms - receive sample to idle system
T=5 ms - receive second sample, send partially filled cell
T=10ms - receive sample to idle system.

Slightly more complex cycles occur for two or three active voices. The probability of "n" active voices has a binomial distribution. This allows the rate at which padding octets are transmitted to be determined.

Filling in the Tables

The model pieces described above allow us to calculate the probability of delay greater than 10 ms, given the line speed and reservations for data and fax. To determine the capacity, we have to guess the number of voice ports and adjust it until we find the number of lines such that "N" lines is less than 1% delay and "N+1" voice ports is more the 1% delay > 10 ms. The capacity for that configuration is "N" voice ports.

Variable Data Bandwidth

In addition to its reserved bandwidth, data can fill in the gaps, if any, in the voice bandwidth. There will be periods where voice completely uses its bandwidth – if these periods didn't occur we would add more voice ports. We would like to know the average bandwidth and the variation about that average.

The average is simple. We know the activity of each voice from the voice model, so we can calculate the bandwidth needed to carry the voice, assuming full ATM cells. The padding calculations can tell us the additional bandwidth consumed by padding when there isn't much voice traffic. Together this is the used voice bandwidth, and the rest is, on average, available for data.

To calculate the variance of this bandwidth, it's easiest to focus on a single conversation. At any point in time it is either active or inactive. It's easy to calculate the transition probabilities of between active and inactive 5 ms later. Every 5 ms that the voice is active, it will generate a voice sample. It's straightforward to build a Markov chain that counts the probability if "n" voice samples in 200 or 1,000 intervals. From these probability distributions, it is easy to calculate the variance of the voice data from a single line in a period of 1 or 5 seconds. The lines are independent, so the variance from n lines is n times as large, and the standard deviation of the group is easy to calculate. Add in the overhead for AAL2, ATM cell headers, and mini-cell headers. This misses the effect of padding - but padding will pull one of the tails of the probability distribution in a bit, so this is a slight overestimate of the variation.

Non-closed Form Results

The system was analyzed for a number of different line rates and reserved data rates (see table 6). The column titled "Average Data Kbps" indicates the average bandwidth available to Internet traffic *when the voice portion of the system is fully loaded*. The actual bandwidth available to Internet traffic will generally be higher because, most of the time, all of the voice lines will not be in use simultaneously. Since the data rate is an average, there is a standard deviation associated with it. The standard deviation varies over time, decreasing as the average is computed over a longer time period. Table 7 gives the standard deviations of the average data rate for 1 sec, 3 sec, 5 sec, and 10 sec, time periods applicable to Internet traffic.

Line Speed Kbps	Simultaneous Calls	Maximum Voice Speed (Kbps)	Reserved Data (Kbps)	Average Data Rate (Kbps)
384	24	320	64	186
	18	256	128	235
	12	192	192	283
512	36	448	64	215
	30	384	128	265
	24	320	192	314
	18	256	256	363
768	63	704	64	249
	56	640	128	307
	50	576	192	356
	43	512	256	414
	30	384	384	521
	18	256	512	619
1000	88	936	64	275
	81	872	128	333
	74	808	192	390
	67	744	256	448
	54	616	384	555
	40	488	512	670
	15	232	768	875
1024	91	960	64	274
	84	896	128	332
	77	832	192	390
	70	768	256	447
	56	640	384	563
	43	512	512	670
	18	256	768	875
1544	148	1480	64	325
	141	1416	128	382
	134	1352	192	440
	127	1288	256	498
	113	1160	384	613
	99	1032	512	729
	71	776	768	959
	46	544	1000	1165
	44	520	1024	1181

Table 6: Results of the non-closed form system simulation.

Note the three lines highlighted in table 6. The first shows that a line operating at only 384Kbps can provide a full T1 of voice channels (24 channels) while simultaneously providing an average of 186Kbps for data. The second shows that a 768Kbps line (half a T1) can provide over two T1's of voice (more than 48 voice channels) while simultaneously providing an average of 350 Kbps of data bandwidth.

The third line shows that a T1 line (1.544 Mbps) can support over four T1's of voice (more than 96 voice channels) while providing an average of almost half a T1 of data access. These are significant advances in local loop, single twisted pair, capacity.

Lines	Standard Deviation in Kbps			
	1 Second	3 Seconds	5 Seconds	10 Seconds
12	23	15	11	8
15	26	16	13	9
18	28	18	14	10
24	33	21	16	12
30	36	23	18	13
36	40	25	20	14
40	42	27	21	15
43	44	27	22	15
50	47	30	23	17
54	49	31	24	17
56	50	31	25	18
63	53	33	26	19
67	54	34	27	19
70	56	25	28	20
74	57	36	28	20
81	60	38	30	21
84	61	38	30	22
88	62	39	31	22
91	63	40	31	23
99	66	42	33	23
113	71	45	35	25
127	75	47	37	27
134	77	49	38	27
141	79	50	39	28
148	81	51	40	29

Table 7: Standard deviation in Kbps of the average data rate available to data, over different time periods. The shaded lines correspond to the shaded lines in the previous table.

Handling fax and modem traffic

None of the speech coders, except G.711, will allow a fax or modem to operate at its maximum data rate. In fact, modems and faxes essentially cannot operate over links which use G.728 and G.729.

One approach is to detect the calling and/or answer tones of a fax or modem call and switch to G.711 (if G.711 is not being used). The problem with this approach is that it requires dynamically reducing the number of voice lines supported, or that certain bandwidth be "set aside" for times when fax or modem calls are done.

A better approach takes advantage of the fact that modems are not commonly used in a business environment, while faxes are. Modems are used for data access and the data portion of the HDSL line will provide this data access. Faxes are common, however, and special care should be given to handling them so that voice capacity is not significantly reduced when they are active.

This can be achieved by using a concept known as "demod-remod" for fax calls. In general, fax machines place both calling and answer tone on the telephone line, making it easy to detect when these calls are in progress. Rather than switching to G.711, these calls can be demodulated on each end, with only the digital data being transmitted across the HDSL link. Since most fax calls operate at 9.6 or 14.4 Kbps, half duplex, these can be carried in the bandwidth of a voice circuit without seriously impacting the voice capacity of the HDSL line.

Data modems can be handled the same way but there is less incentive to utilize data modems when high-speed Internet access is available as part of the communication service. Data modems were not analyzed in our simulation.

Newer fax machines may utilize error control mode (ECM) to improve image quality. With ECM, data is transmitted in blocks with acknowledgments returned to indicate if the blocks were received properly. However, many fax machines, especially older units, do not utilize ECM and simply transmit a page in a continuous stream of bits, most likely at 9.6 or 14.4 Kbps. Since non-ECM mode will have more effect on our system, we chose to focus our analysis on it to determine the worst-case effect of fax.

Thus, we model fax operation as a constant stream of bits, at either 9.6Kbps or 14.4 Kbps, carried by AAL5 cells. In analyzing the system, we continue to maintain the requirement that a minimum amount of bandwidth be allocated to Internet data. The bandwidth for the fax AAL5 cells must come from the bandwidth allocated to voice. Our task now is to determine the number of voice circuits which can be supported when "n" number of voice lines are being used for fax.

We are faced now with a system with three queues: (1) the queue for AAL5 cells with Internet data, (2) the queue for AAL2 cells with voice frames, and (3) the queue for AAL5 cells with fax traffic – see figure 3.

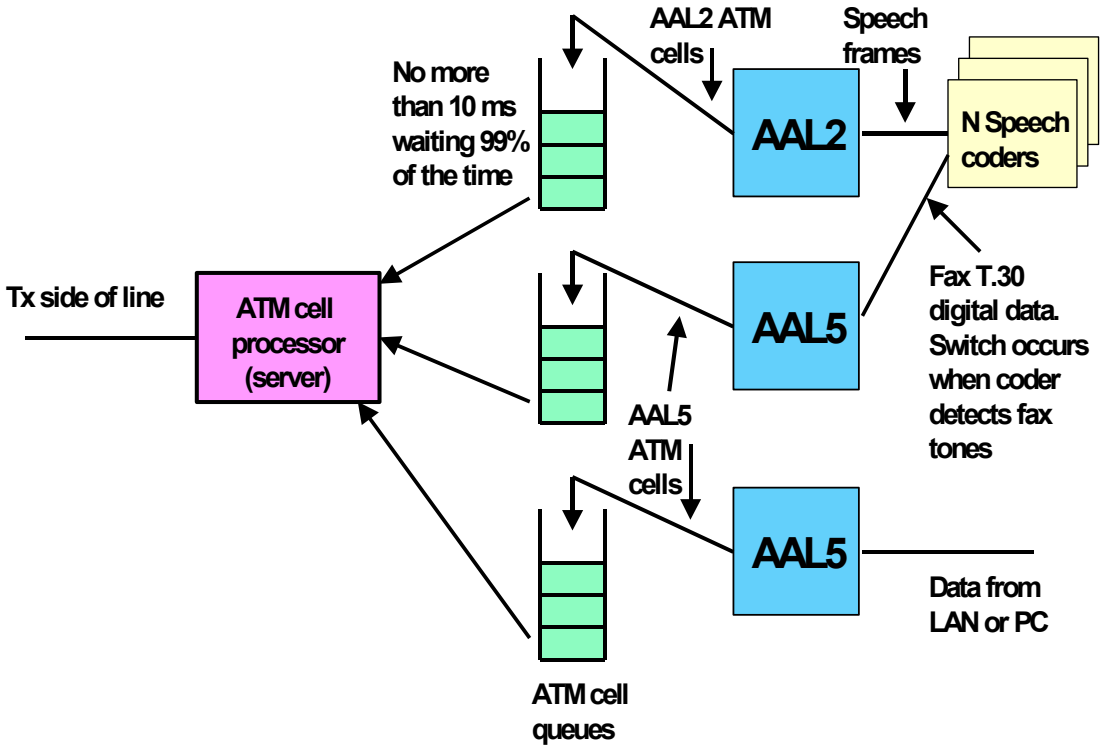

Figure 3: Queueing model when fax demod-remod is added, with transport of T.30 digital fax data by AAL5.

AAL5 adds 8 octets to each protocol data unit (PDU), which may be any size up to 64K-1 octets. To take the worst case, we assumed that each PDU was only 40 octets, meaning that a complete cell of 53 octets was required to transport each 40 octets of fax digital traffic. This means that a 9600 bps fax requires 12,720 bps while a 14,400 bps fax requires 19,080 bps. Longer PDUs could be assumed, which would not cause as great an expansion of the bit rate required to communicate the fax calls.

The problem we face in this situation is to account for the jitter introduced by the fax cells interleaved between the voice cells. This problem was alluded to in the "Jitter" section earlier and will be more fully explored here.

The challenge is to define a scheduler which will provide for sending the voice and fax cells while maintaining the maximum of 10 ms queueing delay, 99% of the time for voice, and avoiding long delays in the fax traffic. Since the fax is not real time, the receive end can provide a buffer equal to the worst-case fax delay.

A simple scheduler can be built from up-down counters, one for each service. Every cell time the desired bandwidth (or a proportional number) is added to each counter. The service with the largest counter gets the cell, and that counter is decremented by the line bandwidth. For some selections of bandwidth, this scheduler will never put a data cell and fax cell back to back - there is at most one "other" cell between two voice cells. The jitter model detects these situations. For higher rates of data and fax, it is possible to get back-to-back data and fax cells, even though the average number of data+fax cells is less than 1.

Line Speed Kbps	Reserved Data Kbps	9600 Fax Calls	14.4K Fax Calls	Simultaneous Calls (including fax calls)	Voice Speed Kbps (excluding fax calls)	Average Data Kbps
768	128	1	0	56	627	302
		2		56	615	298
		3		55	602	301
		5		55	576	292
		10		53	513	286
		0	1	55	621	304
			2	54	602	301
			3	53	583	299
			5	51	545	294
			10	46	449	281

Table 8: Impact of use of some voice lines for fax on the total number of voice lines supported at 768Kbps.

These entries correspond to the line from Table 6 for line speed 768 Kbps and data speed 128 Kbps (which showed 56 voice circuits and an average of 307 Kbps of data). The "Voice Speed Kbps" column is the bandwidth available to the remaining voice circuits, after the appropriate bandwidth required for the fax calls is subtracted (at 12,720 bps for a 9600 bps fax and 19,080 bps for a 14,400 bps fax). The "Simultaneous Calls" column is the total of voice and fax calls – the total number of "voice-grade" lines. Table 6 shows a capacity of 56 voice ports for a 768Kbps line with 128 Kbps guaranteed to voice. This table shows that if 5 ports are used to support 9600 bps fax machines, only 55 voice ports can be used (50 voice and 5 fax), and the average data rate will be reduced to 292 Kbps.

Issues

Several issues arise when speech coders are used for voice. There are a variety of tones which are used by the network, including DTMF for dialing and call progress tones (such as "busy" and "ringing"), including foreign call progress tones when international calling is done. If the voice lines are used as a "trunk" to a PBX, MF tones may be put on the line.

If the lines are used to provision telephony service directly, custom local area signaling services (CLASS) such as Caller-ID, must be passed through.

All of these tones and signals must be detected and passed by the speech coders. G.728 inherently does a better job in this area than lower bit rate coders such as G.729, but special techniques are required for reliable operation.

Finally, the coders must be able to detect fax or modem calls and switch to "demod-remod" to handle these calls. This detection is not easy because a voice telephone may be located in close proximity to a fax machine, causing fax tones to be coupled to the voice line.

These challenges, and more, are being investigated in corporate laboratories today. Most have already been solved, although refinements are still being sought, especially for reliable fax/data modem tone detection.

The Real World

There's always concern about applying the results of a statistical analysis to real world situations – do the statistics we've used really describe the real world? The major statistical item in our analysis is the speech model. The classical speech model has been analyzed over a long period of time and has been shown to adequately represent real speech patterns in the telephone network. However, the telephone network represents a very large number of speakers. When the system consists of a small number of voice circuits, the mean of the speech activity may be larger, and the standard deviation may be significantly greater, than the classical model would predict.

The analysis in this paper has examined the situation when all lines are busy but in the real world, this tends to be an unlikely event, unless the number of lines is very small. For example, suppose the voice circuits are used in a CENTREX type environment, where each line serves an office worker. Because of worker's activities, it is unlikely that everyone will be in their office simultaneously, let alone on the phone. People are out of their office due to sickness, vacation, business travel, in-house business meetings, and even shift work.

And if the voice circuits are used as a trunk for a PBX, the circuits cannot be 100% loaded for very long because of the need to be able to service new calls. The Erlang B or C equations can provide the average busy period usage of trunks, given the blocking probability.

Also, we assumed for our calculations that the data queue is always full and the bandwidth guaranteed to data is unavailable for use by the voice circuits. While this may be true some of the time, it seems unlikely to coincide with maximum utilization of the voice circuits, unless one or more workers are downloading files. After all, if everyone is on the phone, not everyone will be accessing the Internet.

Finally, when analyzing the effect of fax traffic, we assumed non-ECM fax operation. However, almost all modern business fax machines implement ECM. The T.30 digital data of an ECM fax will display periodic burst of traffic, rather than the continuous transmission of data which we assumed in our analysis. Proper analysis of this type of operation would require developing an acceptable model for ECM digital traffic but, intuitively, use of ECM fax machines should improve the link voice capacity compared to our analysis.

These factors tend to make the capacity calculations provided in this paper conservative. The factors specified above can all be modeled and better capacity calculations made, but this is beyond the scope of this paper. Until more exhaustive models are developed, analyzed and published, service providers should be able to use our capacity calculations with confidence that they will be conservative in real world situations.

Summary

This paper has examined the capacity of an HDSL line when used to transport both voice and data, when speech coders and silence suppression are used. The system was analyzed with the standard closed form M/M/1 techniques as well as with simulation. In the simulation model, we attempted to account for as many of the potential delay sources as possible.

Comparing the results of the two techniques, we find that the closed form technique gives results which are optimistic, in terms of number of voice lines supported, compared to the simulation analysis. The

closed form analysis is so much faster and easier, however, being available in an Excel spreadsheet, that it provides a good way to estimate the capacity of a system.

When some of the voice lines are used for fax communications, implementing a demod-remod function will produce less impact on the system capacity than switching to G.711. But even with the demod-remod function, fax usage reduces the number of simultaneous voice circuits which can be supported. When configuring a customer's line, the system designer will have to specify the maximum number of lines which can be used simultaneously for fax transmission.

Overall, the technique of using G.728 speech coders and silence suppression for communicating voice and data over an HDSL line provides a significant increase in capacity compared to traditional techniques. As a rough rule of thumb, this technique can provide four times the number of voice circuits while still providing an average of almost half the line rate for data.

Bibliography

[BRA69] Brady, Paul T., A model for generating on-off speech pattern in two-way conversation. *The Bell System Technical Journal*, Vol 48, pp 2445-2472, Sep 1969.

[DEN95] Deng, Shuang, Traffic characteristics of packet voice. *IEEE International Conference on Communications*. Vol 3, pp 1369-1374, 1995.

[GRO98] Gross, Donald and Carl M. Harris. *Fundamentals of Queueing Theory*. Wiley, 1998.

[NEU95] Neuts, Marcel F., *Matrix-Geometric Solutions in Stochastic Models*. Dover, 1995.