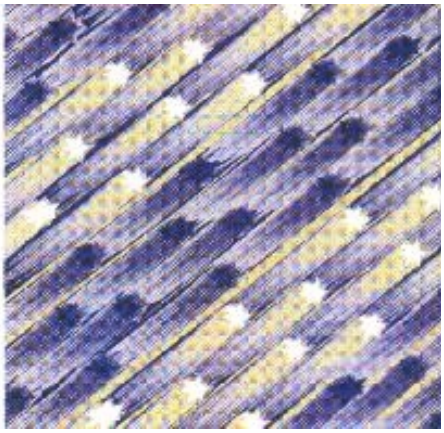

Introduction to ATM Policing

Matt Trzyna

IBM Networking Hardware Division
Product Installation Support
Cary, North Carolina

Abstract

This document describes, in simple terms, how ATM Policing works. Numerous examples and exercises are provided so the reader can “learn by doing”.

Table of Contents

Preface

My first experience with the complexity of ATM Traffic Descriptors and Policing was not a good one.

About 5 years ago, to prepare for an upcoming trade show I had to set up a *simple* ATM PVC to carry eight 1.2mbps MPEG1 video clips from a video server through two Beta Level WAN ATM switches to eight PCs. The Server and PCs were attached to a couple of Campus ATM switches that were each attached by 100mbps ATM TAXI interfaces (Anyone remember those?) to the WAN switches. I had never received any hands on training with ATM equipment, but how hard could it be? I'd already read several books and white papers on ATM that described how circuits should be provisioned and it seemed quite straightforward.

Due to trunking overhead, the biggest PVC I could support over the 45mbps link between the WAN switches was around 42mbps. Doing the math for the actual amount of bandwidth I needed (8 X 1.2mbps = 9.6mbps) this didn't seem to be a problem. I wanted to be *safe*, you know cover the 10% ATM cell overhead and all, so I configured a 35mbps PVC (82,548 cells per second). The LAN ATM switches were also configured for that rate.

Now, if cells were discarded by ejecting them from a slot, I would not have been able to shovel them away fast enough to keep them from burying the WAN switches. Needless to say the video didn't look very smooth. As might be expected my first reaction was: "I need more bandwidth"! I reconfigured the PVC to use the maximum 42mbps but it didn't have any effect. I was now faced with the nightmare of networking: "A performance problem so bad that it can't be solved by throwing bandwidth at it"!

I tried all the different types of traffic descriptor combinations and values: single stage PCR, two stage PCR, PCR+SCR/MBS, with and without tagging. Nothing seemed to help. Not having any traffic analyzers to hook in to see what was actually going on I started playing with the one parameter that I had not yet modified since I really didn't understand it, Cell Delay Variation Tolerance (CDVT). I quickly discovered that if I just used a "big" value (compared to the puny default) the WAN switches stopped discarding cells. I didn't really understand why, but the video looked great and at the time that was all that was important as the truck to carry the equipment to the trade show was scheduled to arrive any minute.

Needless to say "big" CDVT values appeared in all my configurations after that. I tried reading more about how to properly specify traffic descriptors and pick appropriate policing algorithms but the books and papers contained definitions and equations with mathematical symbols like Σ and α but very few examples. I have a feeling the authors of many of those publications really don't get into the lab much (admittedly a snobbish lab rat attitude on my part). Since the publications were of little help, I had to learn through trial and error and by translating the technical jargon and algorithms into some thing I could understand. It wasn't until awhile later

that I discovered the cause of my original problem and why the “big” CDVT made it go away. This will be discussed at the appropriate point later in the document.

The purpose of this paper is to provide as simple and practical explanation as possible of how ATM Policing works. It contains a large number of examples which you can try yourself to make sure you understand. In order to simplify the explanations I’ve taken some *liberties* for which I apologize in advance. Does policing work *exactly* the way I’m going to describe it? The answer is no. With ATM’s many strengths comes a great deal of complexity so to accurately describe its operation you really need things like those $\sum$ and α symbols. Hopefully you’ll get both some information you can put to practical use at once as well as an introduction to the topic which will make the more technical explanations easier to understand.

Though configuring the proper traffic descriptors is also very important you’ve got to start somewhere. Being a “glass is half empty” instead of a “glass is half full” kind of guy, I believe you must first understand why cells are discarded before you can really appreciate how the traffic descriptors should be set. Based on the feedback I receive on this paper, another one titled *Introduction to ATM Traffic Descriptors* would be planned for the future.

The paper is organized as follows:

- Cell Transmission Basics
- UBR/CBR Traffic Policing
- VBR Traffic Policing
- Summary

As this is an introductory document, ABR traffic control will not will not be discussed due to its additional complexity. A basic understanding of ATM is assumed. See the *Recommended Reading* (page 85) if you would like more information.

All mistakes you may find in this document are completely intentional. I’m just trying find out who’s really paying attention so please let me know if you find any.

Please send questions or comments to:

Matt Trzyna
trzyna@us.ibm.com

Policing Basics

In ATM networks there are two basic groups of people, *Service Requesters* and *Service Providers*. The main objective of a *Service Requester* is to obtain the best quality of service (QOS) and most throughput at the lowest cost. The main objective of the *Service Provider* is to support the largest number of *Service Requesters* with as inexpensive an infrastructure as possible. To maximize profit (or just limit expense) *Service Providers* take advantage of the bursty nature of most client/server type traffic by overbooking network resources. The best analogy is the phone system which is not designed so everyone can use the phone at the same time. That's why there are usually problems on Mother's Day.

An ATM switch is not supposed to accept a new connection if the network behind it cannot guarantee the specified QOS (Cell Delay Variation, Cell Delay, Cell Loss, etc.). The amount of resources that must be used in the network to provide the requested QOS is, to a great degree, dependent on the amount of bandwidth allocated for that connection. After a connection is accepted the switch must make sure the requester is not trying to use more bandwidth than was originally specified. *Policing* is the action taken by the ATM switch to ensure incoming traffic does not exceed the rate defined by the traffic descriptors which were agreed on (between the switch and the traffic source) when the connection was established. Let's look at a very high level example.

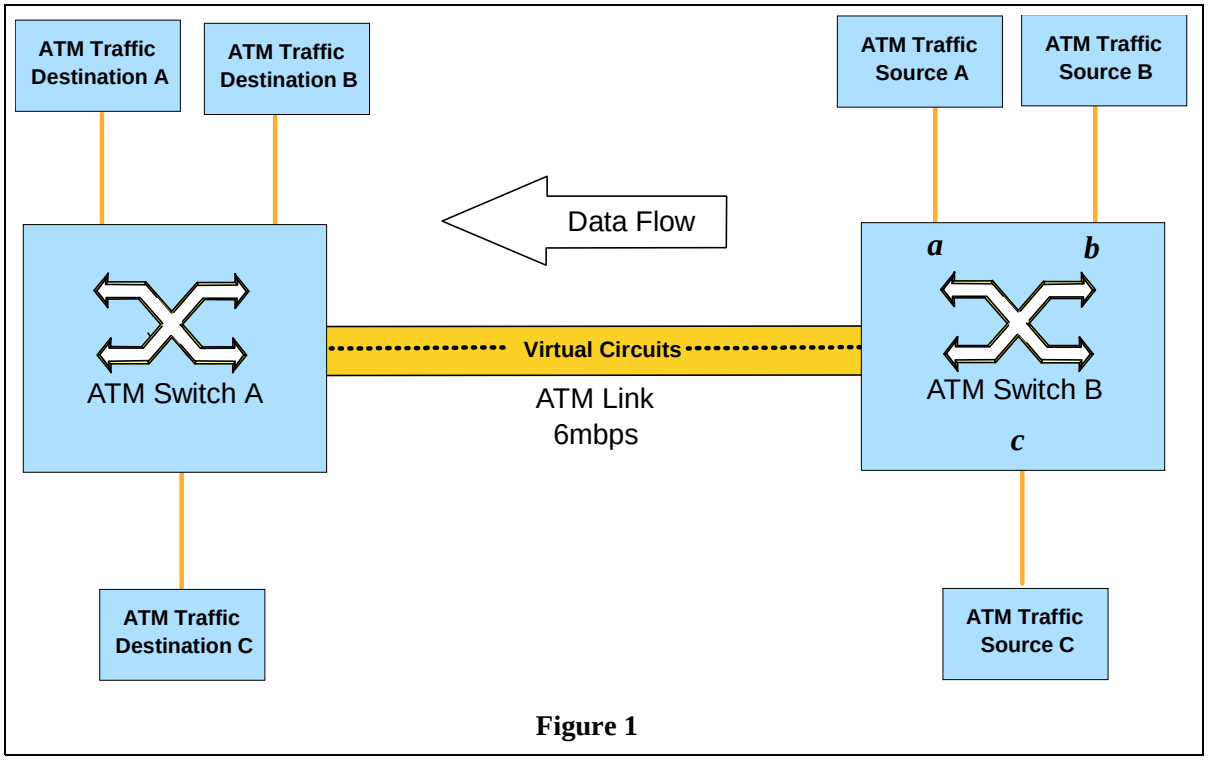

Figure 1

- The accepting or rejecting of a connection is a function titled *Call Admission Control* (CAC).
  - *Source A* requests a 3mbps guaranteed QOS connection to *Destination A*. *Switch B* sees it has 6mbps available on the link to *Switch A* so it establishes the connection. The link now has 3mbps available for other connections.
  - *Source B* now requests a 3mbps guaranteed QOS connection to *Destination B*. *Switch B* sees it has 3mbps available on the link to *Switch A* so it also establishes this connection. The link now has no more bandwidth left for guaranteed QOS connections.
  - *Source C* requests a 2mbps guaranteed QOS connection to *Destination C*. *Switch B* sees it has no bandwidth left to reserve on the link so it rejects the request. If the request had been for a "nonguaranteed QOS" connection ("best effort"), *Switch B* **might** have accepted it. If there was no data being transmitted by *Sources A* and *B* when *Source C* was transmitting it could send the *Source C's* data. If *Source A* and *Source B* were using the link, *Switch B* would discard *Source C's* data but that's ok because there was no guarantee for that traffic. For this example let's just assume *Source C's* connection request was rejected.
- *Switch B* begins **Policing** *Source A's* and *Source B's* traffic (at points ***a*** and ***b*** in the diagram) to ensure they don't use more than their allocated 3mbps. Now, *Station A* starts sending data at a rate greater than 3mbps so its connection is deemed *misbehaving* (I always thought *naughty*, said with a British accent, would have been cuter.). If *Station A* was allowed to send at the higher rate, there may not be bandwidth available on the link when *behaving* (*not naughty*) *Source B* needs it. The goal of ATM congestion control is to keep *misbehaving* connections from impacting *behaving* connections so *Switch B* discards all of *Station A's* traffic that exceeds 3mbps. It could also *tag* the excess traffic instead of discarding it but we'll talk about that later.

Though this may look simple, one of the most difficult things to do in ATM is define a connection's bandwidth parameters (*Traffic Descriptors* in ATM terminology). If these are not accurate one of two things can occur:

- If **more** bandwidth is specified than is actually needed, the data will get through but network resources will be wasted. If the circuit was purchased from a Service Provider you're paying for more bandwidth than you need.
- If **less** bandwidth is specified than is needed, data can be discarded by the switch at ingress which means unhappy users.

As mentioned in the *Preface*, I believe the first step in learning how to specify accurate traffic descriptors is understanding Policing so you know "what you can get away with".

Explanation and Scenario Conventions

Network Overview

All examples and explanations will be based on the following simple network diagram.

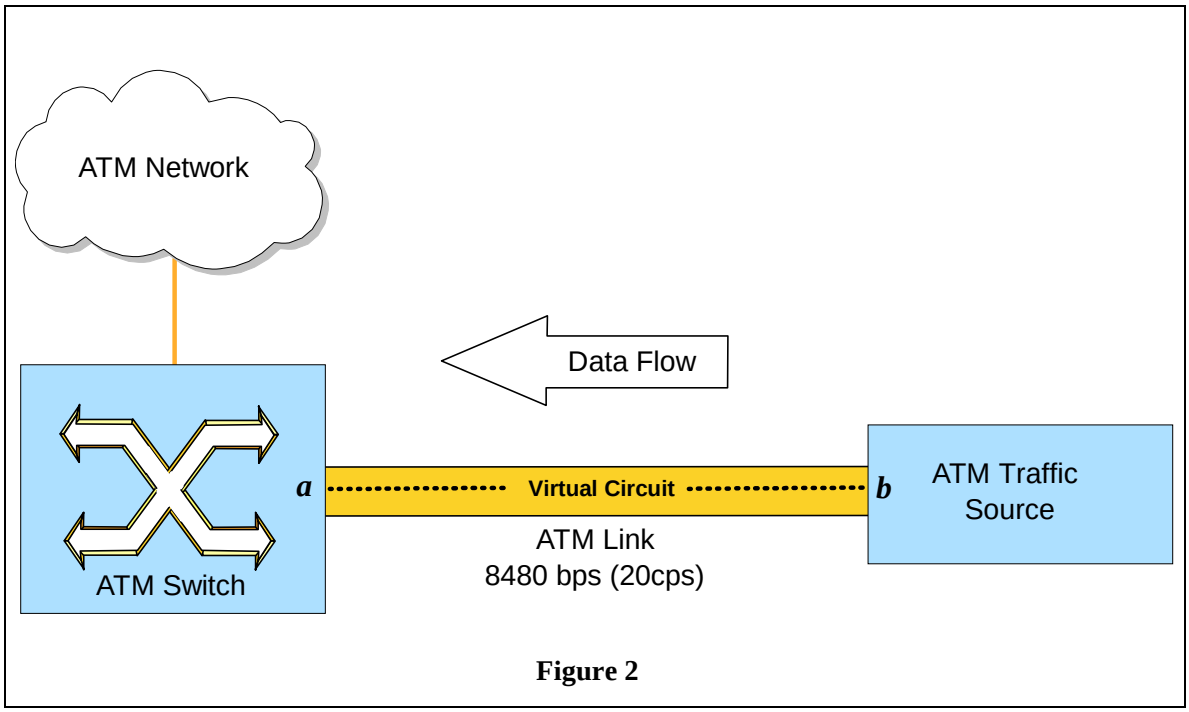

Figure 2

- **ATM Traffic Source**: For the sake of simplicity this hypothetical device will generate the traffic for all the examples. In reality, the combinations of cells in some of the examples could not be generated by a single device. No *Shaping* (explained later) will be performed.
- **ATM Switch**: This will be the device that will be policing the traffic being sent by the ATM Traffic Source. The traffic will be forwarded out into the ***ATM Network*** but that's really not important for our purposes.
- **ATM Link**: One of the problems with understanding ATM traffic descriptors is the size of the numbers involved. I don't know about you but the more numbers there are to the right of a decimal point, the harder it is for me to conceptualize what's going on. To get around this problem we'll be using the new ***MultiMode ATM Tiny Transmission*** interface which has an *Access Rate* (*AR*) of 8480bps. *AR* represents the maximum rate at which data can be transmitted on the interface (Another term for this is *Media Speed*.). To translate the *AR* into *Cells Per Second* (*CPS*) just divide it by the number of bits in an ATM cell (8 * 53 = 424) and you get a whopping **20cps** (8480 / 424)! The ***MATT***

interface is for definitely for people who want to use ATM but don't have a lot to say. As you'll soon see there is no framing; the cells are sent back to back.

- **Data Flow**: Since all the policing will be done by the *ATM Switch* we'll only be concerned with the data flowing *from* point ***b*** in the ATM Traffic Source *to* point ***a*** in the ATM switch where the policing will be done.
- **Virtual Circuit**: Some of the scenarios we'll be covering are unique to Virtual Channel Connections (VCCs) and others to Virtual Path Connections (VPCs). In order to concentrate on the policing operations no distinction will be made. All the examples will just use a generic "circuit"

Access Rate and Idle Cells

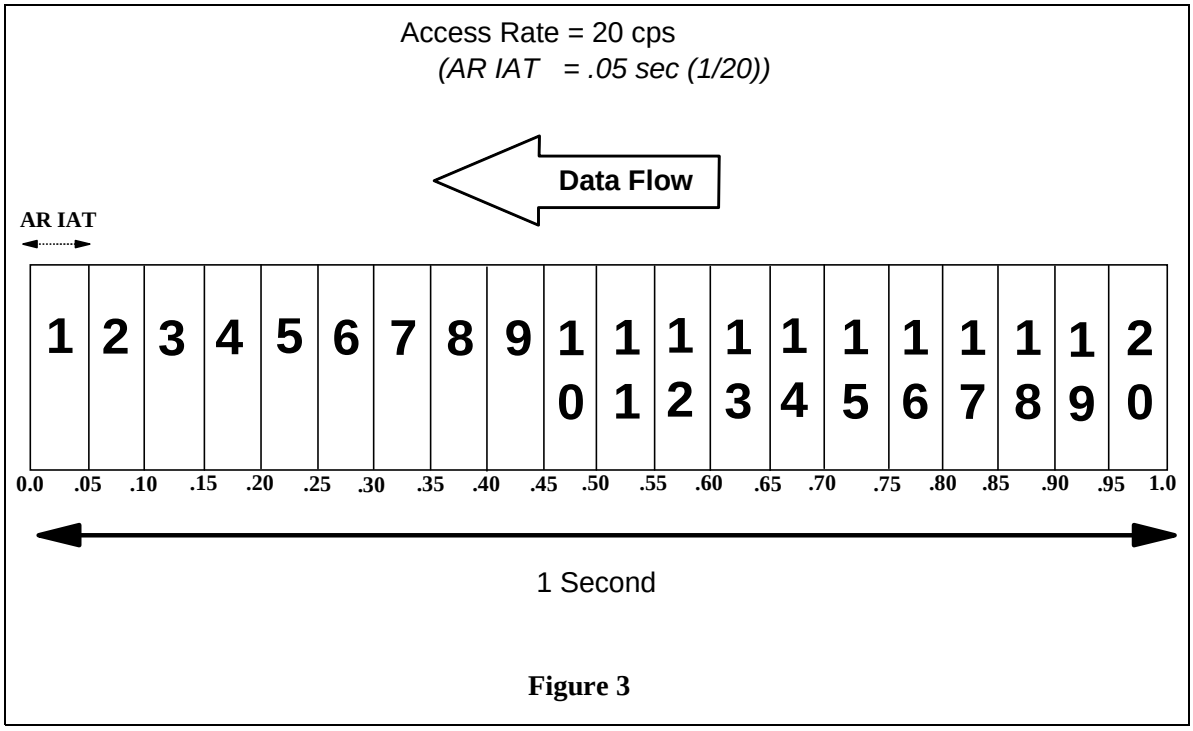

Figure 3

Figure 3 shows the 20cps ***MATT*** interface after the first second of operation. The traffic is flowing from right to left. In *Figure 2* (page 8) this is from point ***b*** in the Traffic Source to point ***a*** in the ATM switch. The diagram is divided into 20 "slots" each containing a number representing a cell received by the ATM Switch (**1** is received first, **2** is received second, etc.).

As explained earlier, Access Rate is the maximum data rate physically possible on an interface. A key ATM traffic concept is cell *Inter-Arrival Time* (*IAT*). *IAT* is the time between start of the receipt of one cell and the start of the receipt of the next. The way to calculate *IAT* is to divide the cell rate into 1 (1/CPS). Thus, for the AR of 20cps the IAT is 1/20 or .05 seconds. As shown along the bottom of the above diagram, a cell is received every .05 seconds.

In order for both ends of an ATM interface to maintain *cell boundary* (knowing when one cell ends and the next begins) there must be a cell sent at every AR IAT. If the sending adapter has no data to send at the *next cell time* (every .05th of a second on the ***MATT*** interface), it inserts an *Idle Cell*. This process is called *Cell Rate Decoupling*. An *Idle Cell* is basically a place keeper which is identified by having VPI 0 and VCI 0 (both reserved values). The receiving adapter simply discards *Idle Cells*. The convention used in the examples will be that the cell numbers for *Idle Cells* will be colored black and displayed in a non-italicized font. I tried really hard to set this up so you wouldn't need to print the document in color. Thus, *Figure 3* shows the interface after one second with only *Idle Cells* having been transmitted.

CBR/UBR Traffic Policing

There are three classes of Policing algorithms based on the type of traffic and/or QOS being provided by the network.

- Constant Bit Rate(CBR)/Unspecified Bit Rate(UBR)
- Variable Bit Rate (VBR)
- Available Bit Rate (ABR)

It's assumed the reader knows the basic differences between these traffic categories. As mentioned in the *Preface* in order to focus on basics, ABR traffic control will not will not be covered due to its complexity. See the *Recommended Readings* (page 85) if you need more information.

This section covers the policing done for CBR and UBR traffic. Only peak bandwidth and cell delay variation requirements are specified.

- In order to achieve the necessary QOS, CBR traffic requires a strong guarantee that the identified bandwidth will be available whenever it's needed. This requires the network to reserve bandwidth at the peak rate at which the circuit can send data. From a network utilization perspective, this is the least efficient type of circuit as it doesn't allow for much *statistical gain* (sharing of the network's bandwidth with other guaranteed connections).
- Though there are no QOS guarantees for UBR traffic, the network must at least know how much traffic is expected. If the network is highly loaded there may be very little excess bandwidth so a UBR connection request could be rejected. Once a UBR connection is accepted the network must ensure it doesn't try to send at a higher rate so there's some degree of fairness toward other UBR connections.

Now let's look at the three different UBR/CBR policing schemes.

- PCR/CDVT (0+1)- No Tagging
- PCR/CDVT (0+1), PCR/CDVT (0)- No Tagging
- PCR/CDVT (0+1), PCR/CDVT (0)- Tagging

One Stage PCR/CDVT Policing - No Tagging

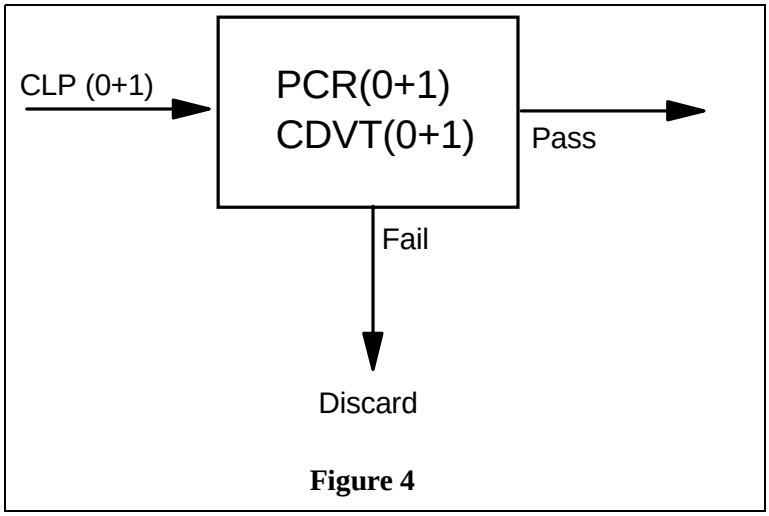

Figure 4

Figure 4 shows what I call the *One Stage* PCR/CDVT algorithm. The *Two Stage* versions will be covered later. Prior to explaining how it works, a few more terms need to be defined.

- **Peak Cell Rate (*PCR*)**: This is the maximum rate, in cells per second, at which a virtual circuit can transmit. The specified value must be less than or equal to the Access Rate.
- **Cell Delay Variation Tolerance (*CDVT*)**: This is one of the more mystical parameters. It's a time value (multiples of the AR IAT) which specifies how much sooner a cell can arrive than what would be expected by the PCR IAT and still be considered conforming. We'll discuss this in more detail later.
- **Cell Loss Priority (*CLP*)**: This is the status of a bit in the cell header which identifies the priority of the cell for discard if there is congestion within the network. The idea is to discard cells with high discard priority (CLP = 1) before discarding any cells with low discard priority (CLP = 0). Some Policing algorithms allow nonconforming cells to be *tagged* (the CLP bit set to 1) instead of being discarded. This is basically saying "I know you were nonconforming but I'll give you a break. I'll let you get through but if there's any congestion in the network, you'll be one of the first to go"! The nomenclature *CLP(0+1)* means "perform the test on all cells regardless of the setting of the CLP bit". Some tests will only be performed on cells that have the bit turned off: *CLP(0)*. Usually the bit is turned on by a policing switch but sometimes an end device which has a UBR connection (no QOS guarantees) can turn the bit on before it transmits the cell to indicate "Hey, I'm on a UBR connection, throw me away first if you have to"!

Though a flowchart for *One Stage PCR/CDVT* Policing is quite simple, later *Two Stage* algorithms, especially when Sustained Cell Rate is involved, are very complex. Since the objective of this paper is to make policing understandable, I'll explain the operation via examples rather than flowcharts or pseudo code.

One Stage PCR/CDVT Policing: Example 1

We'll start with an explanation of the policing of the following one second cell stream.

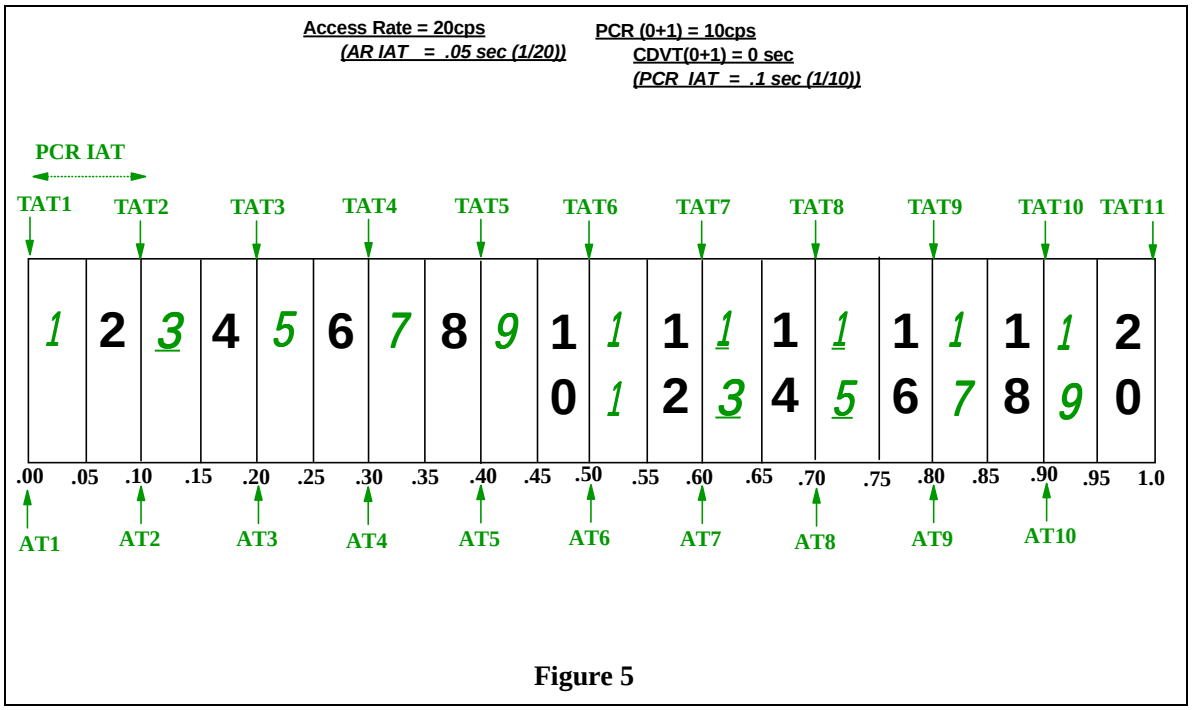

Figure 5

Several of the new terms and diagram conventions which appear in the above figure need to be defined before we start. Don't worry about understanding all of them now. We'll be going through each example step by step so after awhile it will become clear.

.

- **Arrival Time *(AT)***: This will be used to signify the arrival times of conforming and nonconforming cells.
- **PCR IAT**: This is the PCR Inter-Arrival Time. Just like the AR IAT it's calculated by dividing the PCR value into 1 (1/10 = .1sec). This is the value which is used to determine the soonest the next conforming cell at the PCR rate can arrive.
- **Theoretical Arrival Time *(TAT)***: TAT is a time value which indicates the soonest the **next** cell can arrive and still be considered conforming. A new TAT is only calculated based on the arrival of a conforming data cell (Idle Cells are ignored). If the data cell was not conforming the TAT is not changed. Now, how the expected TAT for the next conforming cell is calculated depends on the scenario. This will be explained a bit closer to when you'll need to understand it to do the exercises.

With regard to diagram conventions:

- *IAT Arrows*: Arrows pointing to the arrival times of data cells will be along the bottom of the diagram.
- *TAT Arrows*: Arrow pointing to the TATs will be along the top of the diagram.

- Cell Types
  - Idle Cells will be identified by black un-italicized numbers: **1, 2, 3,** etc.
  - Data Cells will be identified by italicized numbers:
    - Conforming cells:
      - Their cell numbers will be green: *1, 2, 3,* etc.
      - The AT arrows will also be colored green and numbered 1, 2, 3, etc. to indicate the first, second, third, etc. conforming cell received.
    - Non Conforming cells:
      - Their cell numbers will be colored red: *1, 2, 3,* etc.. The cell spaces will also be crossed out to emphasize deletion.
      - The AT arrows will also be colored red and won't be numbered since, as you'll see, they don't factor into later conformance calculations. An X will be placed where the number would have gone.
    - Mystery Cells: These will be data cells for which I'll ask **you** to determine if they are conforming or not.
      - Their cell numbers will be colored blue: *1, 2, 3,* etc.
      - The AT arrows will also be colored blue. The numbers will be left blank for you to fill in, or not, as appropriate.
    - Tagged Cells (cells that have CLP=1) will be indicated by underlined cell numbers: *1*, *2*, *3*, etc.

Now, how do you determine if a cell is conforming?

- A data cell is determined to be conforming when it is received in the time frame indicated by ***TATn-CDVT*** where *TATn* represents the TAT for the next expected data cell. Don't worry about *CDVT* for now. It complicates things a bit so for these initial exercised it will be set to 0sec and won't have any effect.
- A new TAT is only calculated as follows based on the arrival of a **conforming data cell** (Idle Cells are ignored).
  - If the conforming data cell arrived at a time that is **not later than its TAT** the next TAT is calculated by: *Current TAT + PCR IAT*. "*Not later than its TAT*" is specified instead of "*equal to its TAT*" due to implications of CDVT which will be covered later.
  - If the conforming data cell arrived at a time that is **later than its TAT** the next TAT is calculated by: *Data Cell's Arrival Time + PCR IAT*.

I know what you're thinking. "He said this was going to be simple and now there are all of these conventions to keep track of". Trust me, as you start going through the examples it gets much easier.

Now let's walk through the example in *Figure 5* (page 13) to see how the policing operates. Note that the PCR is set to 10cps which is half of the AR. If you have this document stapled together

now is a good time to take the staple out so you can keep the diagrams next to the explanations to save you from having to flip back and forth.

- Cell 1 is a CLP(0) data cell (it's not underlined) and is received at time .00sec. This is the first cell received so it has to be conforming.
  - *AT1* is set to .00sec.
  - Since *theoretically* the earliest any data cell could have arrived would be at time .00 we'll always call it TAT1.
  - TAT2 = .1sec.: AT1 was not later than TAT1 so TAT2 = TAT1 + PCR IAT. **Note that this is the earliest the next data cell can arrive and still be conforming.**
- Cell 2 is an Idle Cell. It's just used as a place holder.
- Cell 3 is a CLP(1) data cell (underlined). Since it was not received before TAT2-CDVT it is conforming.
  - AT2 is set to .1sec. (the arrival time of this cell)
  - TAT3 = .2sec.: AT2 was not later than TAT2 so TAT3 = TAT2 + PCR IAT.
- Cell 4 is an Idle Cell.
- Cell 5 is a CLP(0) data cell (not underlined). Since it was not received before TAT3-CDVT it is conforming.
  - AT3 is set to .2sec.
  - TAT4 = .3sec.: AT3 was not later than TAT3 so TAT4 = TAT3 + PCR IAT.
- Cell 6 is an Idle Cell.
- Cell 7 is a CLP(0) data cell. Since it was not received before TAT4-CDVT it is conforming.
  - AT4 is set to .3sec.
  - TAT5 = .4sec.: AT4 was not later than TAT4 so TAT5 = TAT4 + PCR IAT.
- Cell 8 is an Idle Cell.
- Cell 9 is a CLP(0) data cell. Since it was not received before TAT5-CDVT it is conforming..
  - AT5 is set to .4sec.
  - TAT6 = .5sec.: AT5 was not later than TAT5 so TAT6 = TAT5 + PCR IAT.
- Cell 10 is an Idle Cell.
- Cell 11 is a CLP(0) data cell. Since it was not received before TAT6-CDVT it is conforming..
  - AT6 is set to .5sec.
  - TAT7 = .6sec.: AT6 was not later than TAT6 so TAT7 = TAT6 + PCR IAT.
- Cell 12 is an Idle Cell.
- Cell 13 is a CLP(1) data cell. Since it was not received before TAT7-CDVT it is conforming.
  - AT7 is set to .6sec.
  - TAT8 = .7sec.: AT7 was not later than TAT7 so TAT8 = TAT7 + PCR IAT.
- Cell 14 is an Idle Cell.
- Cell 15 is a CLP(1) data cell. Since it was not received before TAT8-CDVT it is conforming.
  - AT8 is set to .7sec.
  - TAT9 = .8sec.: AT8 was not later than TAT8 so TAT9 = TAT8 + PCR IAT.

- Cell 16 is an Idle Cell.
- Cell 17 is a CLP(0) data cell. Since it was not received before TAT9-CDVT it is conforming.
  - AT9 is set to .8sec.
  - TAT10 = .9sec.: AT9 was not later than TAT9 so TAT10 = TAT9 + PCR IAT.
- Cell 18 is an Idle Cell.
- Cell 19 is a CLP(0) data cell. Since it was not received before TAT10-CDVT it is conforming.
  - AT10 is set to .9sec.
  - TAT11 = 1sec.: AT10 was not later than TAT10 so TAT11 = TAT10 + PCR IAT.
- Cell 20 is an Idle Cell.

I wanted to start out on a positive note so all the data cells in this example were conforming and the full PCR rate of 10cps was used. Take note of a few things:

- The tagged (CLP=1) and un-tagged (CLP=0) cells were all treated the same.
- The Idle Cells were basically ignored. They only act as place holders.

Next is a slightly different scenario. Stick with it. Believe me, after awhile you'll be able to do all this in your head (though I'll keep being explicit in the cell by cell explanations).

One Stage PCR/CDVT Policing: Example 2

Note that the PCR has been reduced to 5cps which now makes the PCR IAT = .2sec.

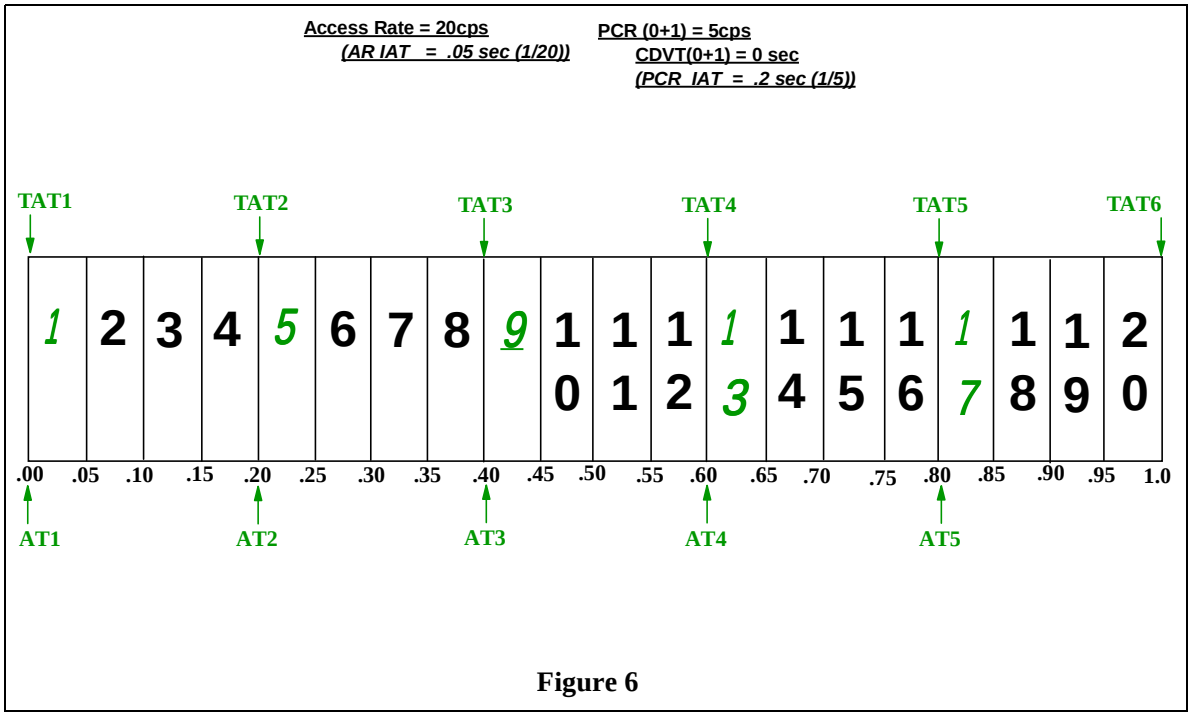

Figure 6

- Cell 1 is a CLP(0) data cell and is received at time .00sec. This is the first cell received so it has to be conforming.
  - *AT1* is set to .00sec.
  - TAT2 = .2sec.: AT1 was not later than TAT1 so TAT2 = TAT1 + PCR IAT.
- Cells 2, 3, and 4 are Idle Cells.
- Cell 5 is a CLP(0) data cell. Since it was not received before TAT2-CDVT it is conforming.
  - AT2 is set to .2sec.
  - TAT3 = .4sec.: AT2 was not later than TAT2 so TAT3 = TAT2 + PCR IAT.
- Cells 6, 7, and 8 are Idle Cells.
- Cell 9 is a CLP(1) data cell. Since it was not received before TAT3-CDVT it is conforming.
  - AT3 is set to .4sec.
  - TAT4 = .6sec.: AT3 was not later than TAT3 so TAT4 = TAT3 + PCR IAT.
- Cells 10, 11, and 12 are Idle Cells.
- Cell 13 is a CLP(0) data cell. Since it was not received before TAT4-CDVT it is conforming.
  - AT4 is set to .6sec.
  - TAT5 = .8sec.: AT4 was not later than TAT4 so TAT5 = TAT4 + PCR IAT.
- Cells 14, 15, and 16 are Idle Cells.

- Cell 17 is a CLP(0) data cell. Since it was not received before TAT5-CDVT it is conforming.
  - AT5 is set to .8sec.
  - TAT6 = 1sec.: AT5 was not later than TAT5 so TAT6 = TAT5 + PCR IAT.
- Cells 18, 19, and 20 are Idle Cells.

Again, all the data is conforming at the full PCR rate. Notice how the same process is followed even though the PCR value changed.

Now, try one on your own.

One Stage PCR/CDVT Policing: Example 3

The PCR is the same in as the previous example but the arrival times of the data cells are different. Mark up the diagram to show the TATs and ATs.

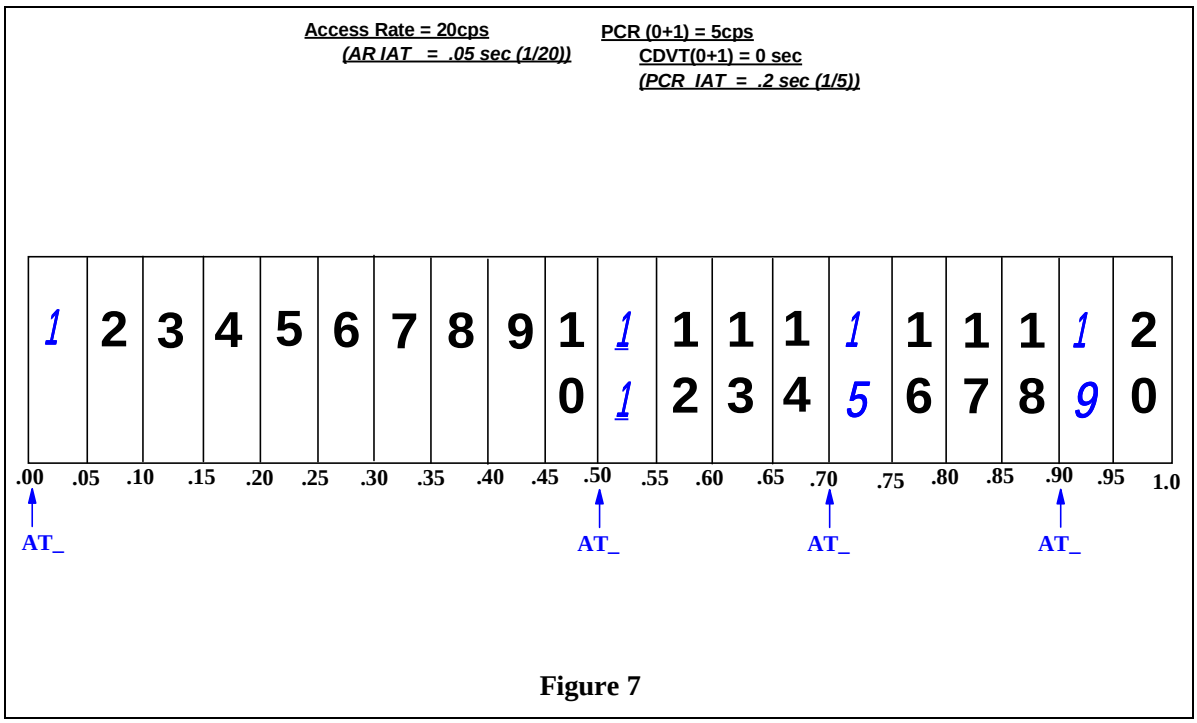

Figure 7

This exercise really wasn't fair since I used it to introduce a new point that might have caused some confusion.

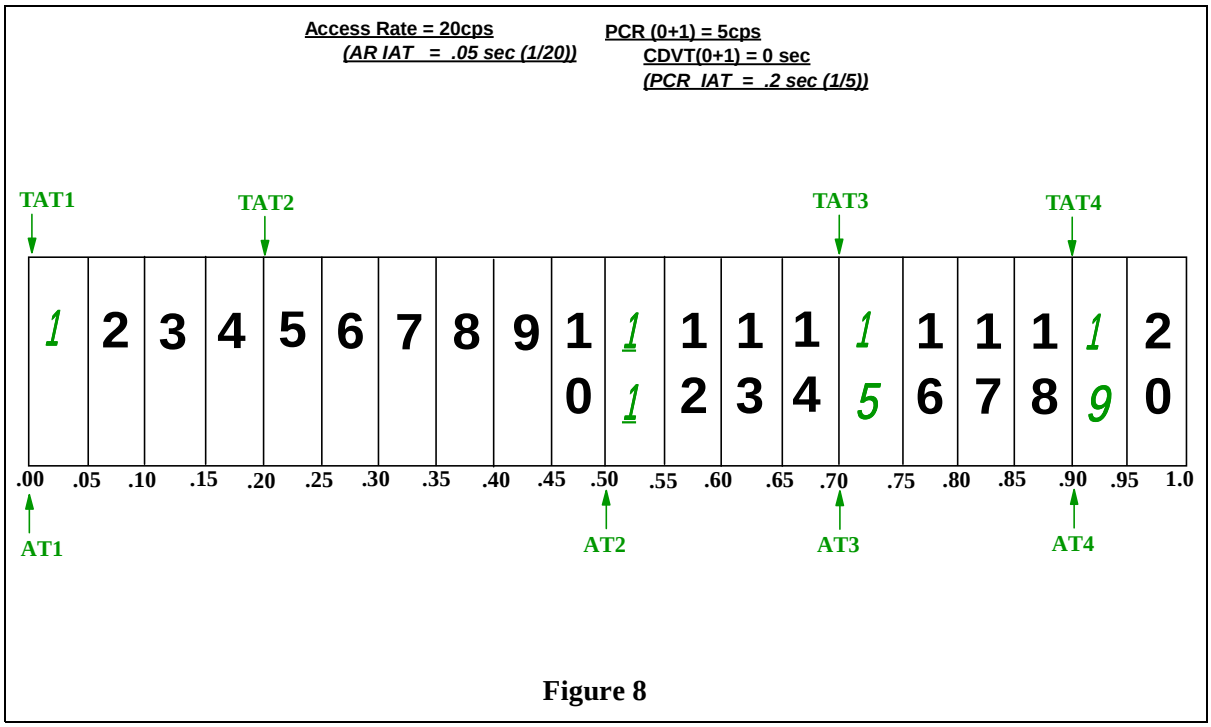

Figure 8

- Cell 1 is a CLP(0) data cell and is received at time .00sec. This is the first cell received so it has to be conforming.
  - *AT1* is set to .00sec.
  - TAT2 = .2sec.: AT1 was not later than TAT1 so TAT2 = TAT1 + PCR IAT.
- Cells 2 through 10 are Idle Cells.
- Cell 11 is a CLP(1) data cell. Since it was not received before TAT2-CDVT it is conforming.
  - AT2 is set to .5sec.
  - TAT3 = .7sec.: AT2 is **later** than TAT2 so TAT3 = **AT2** + PCR IAT. **Please note that something different was done here than you've seen before.** I did warn you about this situation before we started the examples but I bet you didn't remember. Since Cell 11 arrived **after** the expected TAT, the next TAT is calculated using the cell's Arrival Time (AT) instead of the last expected TAT (TAT2). You can see what would have happened if we had used TAT2: The expected arrival time of the next cell would have been *earlier* than the time Cell 11 was actually received! Obviously this wouldn't work until some sort of *ATM Time Travel* function is approved by the ATM Forum.
- Cell 12, 13, and 14 are Idle Cells.
- Cell 15 is a CLP(0) data cell. Since it was not received before TAT3-CDVT it is conforming.
  - AT3 is set to .7sec.
  - TAT4 = .9sec.: AT3 was not later than TAT3 so TAT4 = TAT3 + PCR IAT. **Note we used the last TAT3 as the increment here. About now you may be wondering**

why not just ALWAYS used the AT to increment. When we get to CDVT scenarios where the AT is earlier than the TAT but the cell is still conforming, you'll understand.

- Cell 16, 17, and 18 are Idle Cells.
- Cell 19 is a CLP(0) data cell. Since it was not received before TAT4-CDVT it is conforming.
  - AT4 is set to .9sec.
  - TAT5 = 1.1sec.: AT4 was not later than TAT4 so TAT5 = TAT4 + PCR IAT. (off the chart)
- Cell 20 is an Idle Cell.

Again, all the data cells were conforming and you learned that arriving *later* than expected is ok in the ATM Police state. Now let's see what a *misbehaving* user is like.

One Stage PCR Policing: Example 4

Figure out the ATs and TATs and identify which cells are conforming and which are nonconforming. I know that I haven't provided an example of nonconformance yet but if you just followed the procedures used before you should be able to figure it out. Anyway, what's the sport in explaining it first?

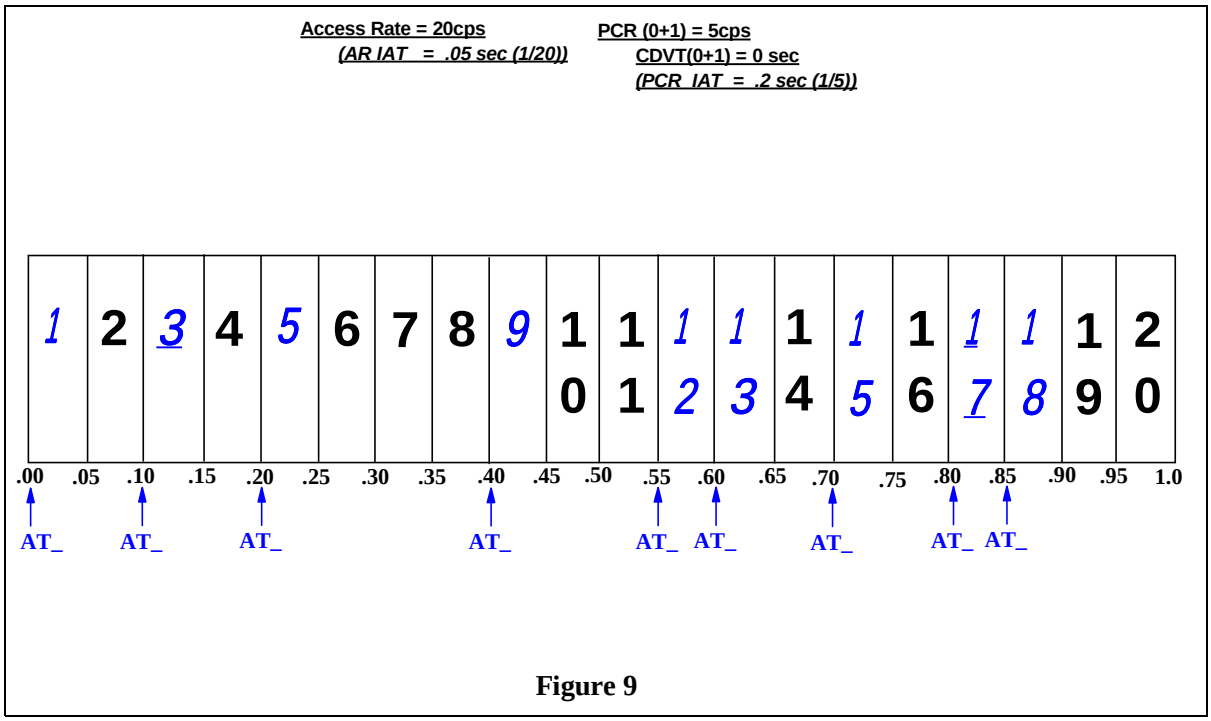

Figure 9

Now don't you feel better having tried to figured it out by yourself? Shame on you if you just looked ahead for the answer!

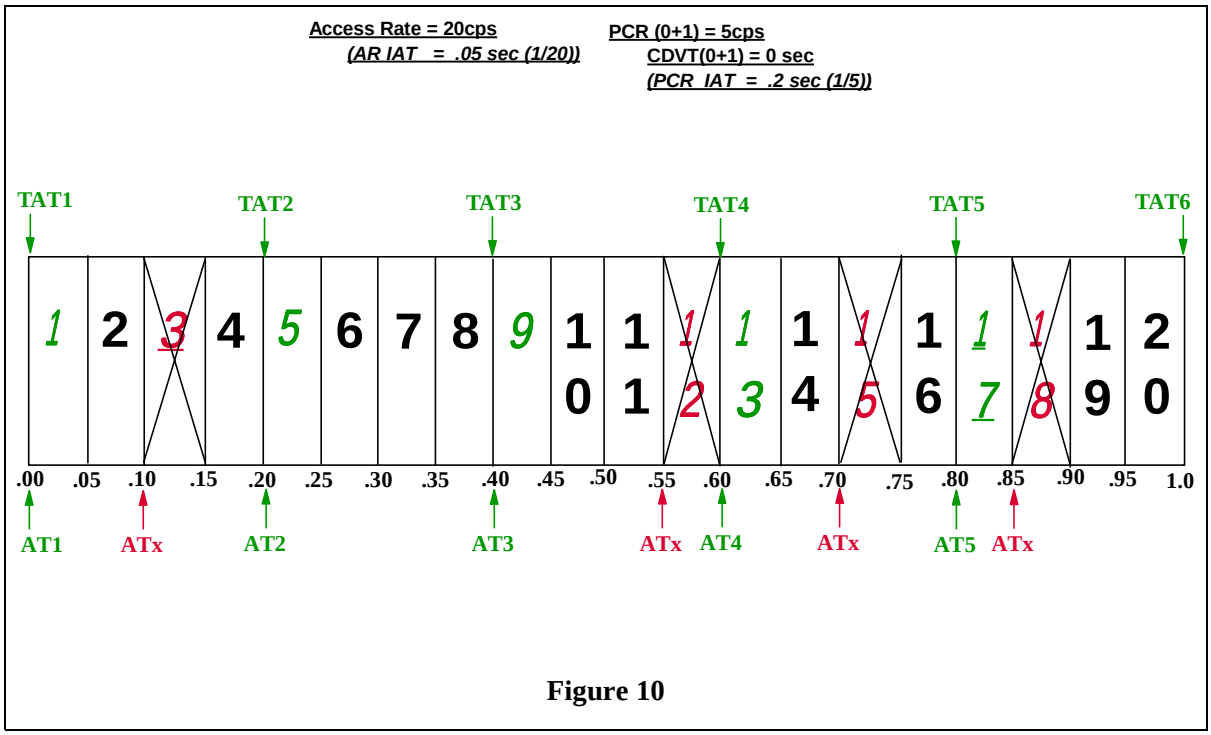

Figure 10

- Cell 1 is a CLP(0) data cell and is received at time .00sec. This is the first cell received so it has to be conforming.
  - *AT1* is set to .00sec.
  - TAT2 = .2sec.: AT1 was not later than TAT1 so TAT2 = TAT1 + PCR IAT.
- Cells 2 is an Idle Cell.
- Cell 3 is a CLP(1) data cell. Since it **was** received **before** TAT2-CDVT it is nonconforming (a *naughty* cell) and will be discarded.
  - AT is **not** reset.
  - TAT is **not** reset.
- Cell 4 is an Idle Cell.
- Cell 5 is a CLP(0) data cell. Since it was not received before TAT2-CDVT it is conforming.
  - AT2 is set to .2sec.
  - TAT3 = .4sec.: AT2 was not later than TAT2 so TAT3 = TAT2 + PCR IAT.
- Cells 6, 7, and 8 are Idle Cells.
- Cell 9 is a CLP(0) data cell. Since it was not received before TAT3-CDVT it is conforming.
  - AT3 is set to .4sec.
  - TAT4 = .6sec.: AT3 was not later than TAT3 so TAT4 = TAT3 + PCR IAT.
- Cells 10 and 11 are Idle Cells.
- Cell 12 is a CLP(0) data cell. Since it **was** received **before** TAT4-CDVT it is nonconforming (another *naughty* cell) and will be discarded.
  - AT is **not** reset.
  - TAT is **not** reset.

- Cell 13 is a CLP(0) data cell. Since it was not received before TAT4-CDVT it is conforming.
  - AT4 is set to .6sec.
  - TAT5 = .8sec.: AT4 was not later than TAT4 so TAT5 = TAT4 + PCR IAT.
- Cell 14 is an Idle Cell.
- Cell 15 is a CLP(0) data cell. Since it **was** received **before** TAT5-CDVT it is nonconforming and will be discarded.
  - AT is **not** reset.
  - TAT is **not** reset.
- Cell 16 is an Idle Cell.
- Cell 17 is a CLP(1) data cell. Since it was not received before TAT5-CDVT it is conforming.
  - AT5 is set to .8sec.
  - TAT6 = 1.0sec.: AT5 was not later than TAT5 so TAT6 = TAT5 + PCR IAT.
- Cell 18 is a CLP(0) data cell. Since it **was** received **before** TAT6-CDVT it is nonconforming and will be discarded.
  - AT is **not** reset.
  - TAT is **not** reset.
- Cells 19 and 20 are Idle Cells.

So this *naughty* user only paid for a 5cps circuit but tried to run at 9cps (probably the same type of person who takes up two parking spots so their car won't get scratched). Thanks to policing, we enforced the 5cps data rate and avoided possibly catastrophic congestion in the network.

Now that you're hopefully a bit more comfortable with AT, IAT, and TAT let's complicate matters and really start using CDVT.

CDVT Overview

As explained earlier, Cell Delay Variation Tolerance is a time value (multiples of the AR IAT) which specifies how much sooner a cell can arrive than what would be allowed by the PCR and still be considered conforming. It's like good-cop/bad-cop.

- PCR: "You think you're above the law, cell? You arrived earlier than the TAT so your just going to have to be ***discarded*** "!
- CDVT: "Take it easy on the kid, PCR. He was only a little bit early. Let's let him pass."

Actually, the reason the cells are allowed to come in early is that there are situations in which the sending station was transmitting at a conforming rate but things occurred in the network which were beyond the its control that caused the problem. For example:

- Congestion in the network can cause the cell stream (which had earlier entered at a conforming rate) to be buffered in an interim switch. By the time the stream finally gets its turn for transmission, a bunch of the cells had been queued and were burst out at a

faster rate than they entered. The formal term for this situation is *Cell Clumping*. You would have thought the people who came up with terms like *Theoretical Arrival Time* and *Cell Delay Variation Tolerance* could have come up with something a little more sophisticated sounding. A policing switch at the egress point would discard the cells sent at the higher rate.

- Multiplexing VCCs into a VPC can result in the a situation similar to the one described above.

A function called *Shaping* addresses most of the causes of *Cell Clumping*. When performing *Shaping*, the transmitting device buffers egress traffic and feeds it out to the policing switch at the agreed on rate. Unfortunately, *Shaping* is not a required function so it may not be possible to use it in your situation.

CDVT is **not** considered a source traffic parameter since in most cases the source has little if any control over the cause of the problem. Because if this it is usually not specified by the user but in the network's switch.

In addition to addressing *Cell Clumping* I've also found CDVT useful as a "fine tuning" mechanism when there are excessive discards even when the traffic descriptors (for *Policing* and *Shaping*) match at both ends of the connection. Though there are standardized algorithms for *Policing* and *Shaping*, each vendor's implementation could be different enough so that Vendor A's shaped traffic may still look too high to Vendor B's policing switch.

Now that I've probably confused most of you and upset the rest with my over generalizations, let's look at some examples of how CDVT is used.

One Stage PCR/CDVT Policing: Example 5

The PCR is 5cps which means an IAT of .2 seconds (4 AR cell times). CDVT indicates the number of AR cell times a data cell can be received **earlier** than TAT and still be conforming. Thus, a .1sec CDVT means a data cell can arrive up to 2 AR Cell times earlier than the TAT.

If you feel lucky (*Well, ... do you?*) you can try it yourself first. The rest of you can just move directly to the cell by cell explanation.

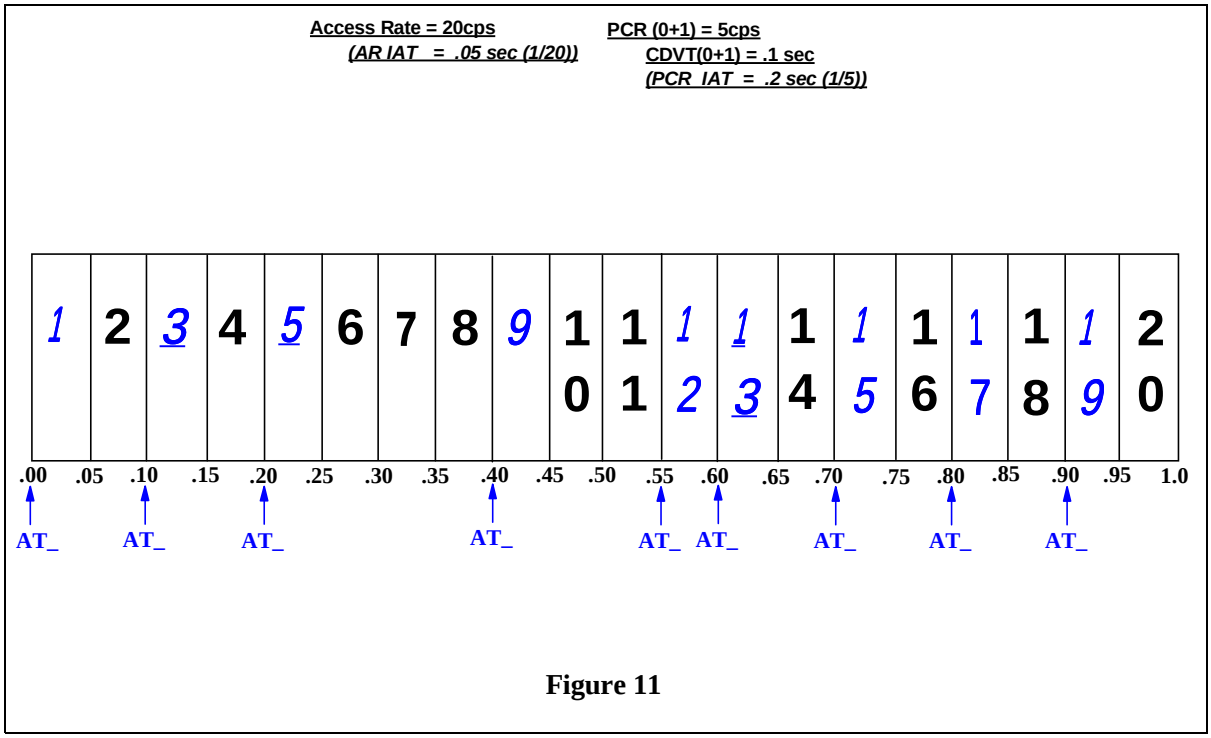

Figure 11

The answer is on the next page.

Here's how it turns out. The range of the CDVT in relation to the TAT is indicated by lines above the cells (remember left is earlier).

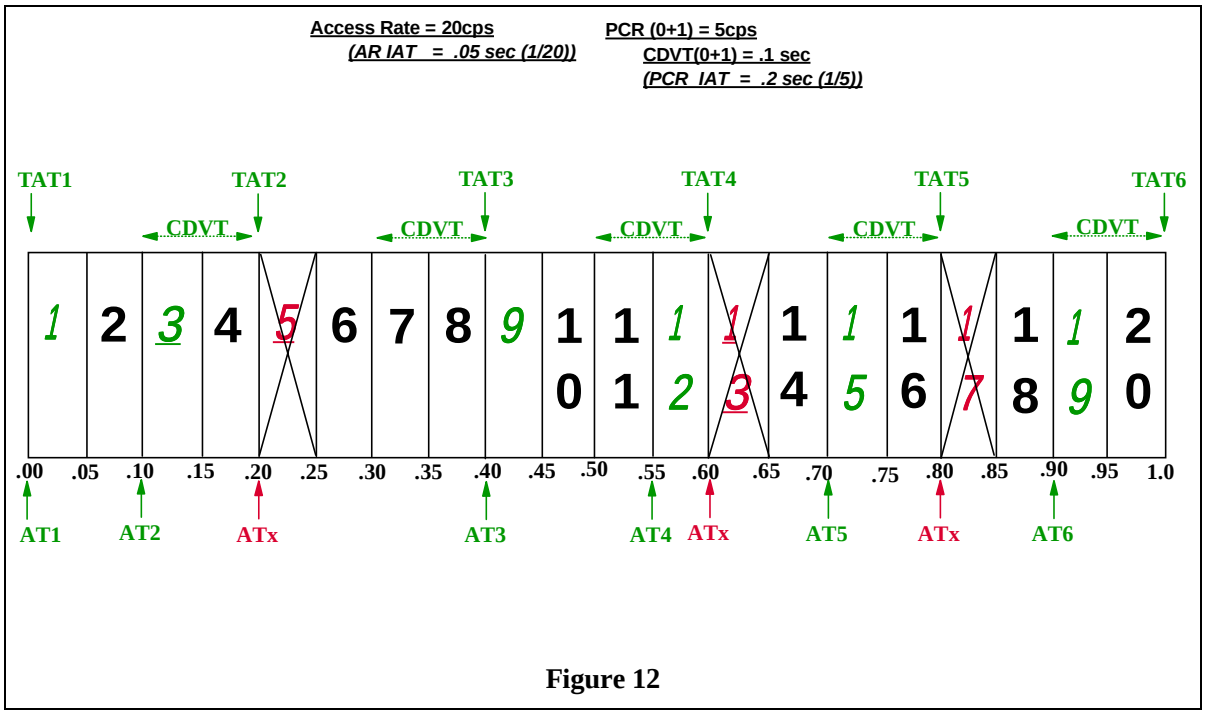

Figure 12

- Cell 1 is a CLP(0) data cell and is received at time .00sec. This is the first cell received so it has to be conforming.
  - *AT1* is set to .00sec.
  - TAT2 = .2sec.: AT1 was not later than TAT1 so TAT2 = TAT1 + PCR IAT.
- Cell 2 is an Idle Cell.
- Cell 3 is a CLP(1) data cell. Since it was not received before TAT2-CDVT it is conforming. **Note it was received before TAT2 but within the range of TAT2-CDVT (.2-.1). If CDVT had been set to .0sec or .05sec this cell would have been considered naughty.**
  - AT2 is set to .1sec.
  - TAT3 = .4sec.: AT2 was not later than TAT2 so TAT3 = TAT2 + PCR IAT. **This is why you can't use the AT to calculate the next TAT. If this was allowed the TATs would keep shifting to the left due to the CDVT. You can see if this was done MANY more cells would be considered conforming than allowed for by the specified PCR.**
- Cell 4 is an Idle Cell.
- Cell 5 is a CLP(1) data cell. Since it was received **before** TAT3-CDVT it is nonconforming and will be discarded. **Remember TAT3 is the current *next TAT* , not TAT2.**
  - AT is not reset.
  - TAT is not reset.
- Cells 6, 7, and 8 are Idle Cells.

- Cell 9 is a CLP(0) data cell. Since it was not received before TAT3-CDVT it is conforming.
  - AT3 is set to .4sec.
  - TAT4 = .6sec.: AT3 was not later than TAT3 so TAT4 = TAT3 + PCR IAT.
- Cells 10 and 11 are Idle Cells.
- Cell 12 is a CLP(0) data cell. Since it was not received before TAT4-CDVT it is conforming. **Saved again by the CDVT.**
  - AT4 is set to .55sec.
  - TAT5 = .8sec.: AT4 was not later than TAT4 so TAT5 = TAT4 + PCR IAT.
- Cell 13 is a CLP(1) data cell. Since it was received **before** TAT5-CDVT it is nonconforming and will be discarded. **Remember TAT5 is the current *next TAT*.**
  - AT is not reset.
  - TAT is not reset.
- Cell 14 is an Idle Cell.
- Cell 15 is a CLP(0) data cell. Since it was not received before TAT5-CDVT it is conforming. **Saved again by the CDVT.**
  - AT5 is set to .7sec.
  - TAT6 = 1.0sec.: AT5 was not later than TAT5 so TAT6 = TAT5 + PCR IAT.
- Cell 16 is an Idle Cell.
- Cell 17 is a CLP(0) data cell. Since it was received **before** TAT6-CDVT it is nonconforming and will be discarded.
  - AT is not reset.
  - TAT is not reset.
- Cell 18 is an Idle Cell.
- Cell 19 is a CLP(0) data cell. Since it was not received before TAT6-CDVT it is conforming. **Saved again by the CDVT.**
  - AT6 is set to .9sec.
  - TAT7 = 1.2sec.: AT6 was not later than TAT6 so TAT7 = TAT6 + PCR IAT (off the chart).
- Cells 20 is an Idle Cell.

Lots of *naughty* cells but did you notice something else? If you count them up 9 data cells were sent and 6 were conforming even though the PCR was 5! Thus, depending on its size, CDVT can result in more conforming cells getting through than the PCR would seem to allow. To be fair though, due to the low rate of the ***MATT*** interface the CDVT of .1sec is really 50% of the PCR IAT which is really quite large.

Now let's see if you can do a CDVT scenario yourself.

One Stage PCR/CDVT Policing: Example 7

Identify the conforming and nonconforming cells in the following scenario. The PCR and CDVT are the same as in the previous example.

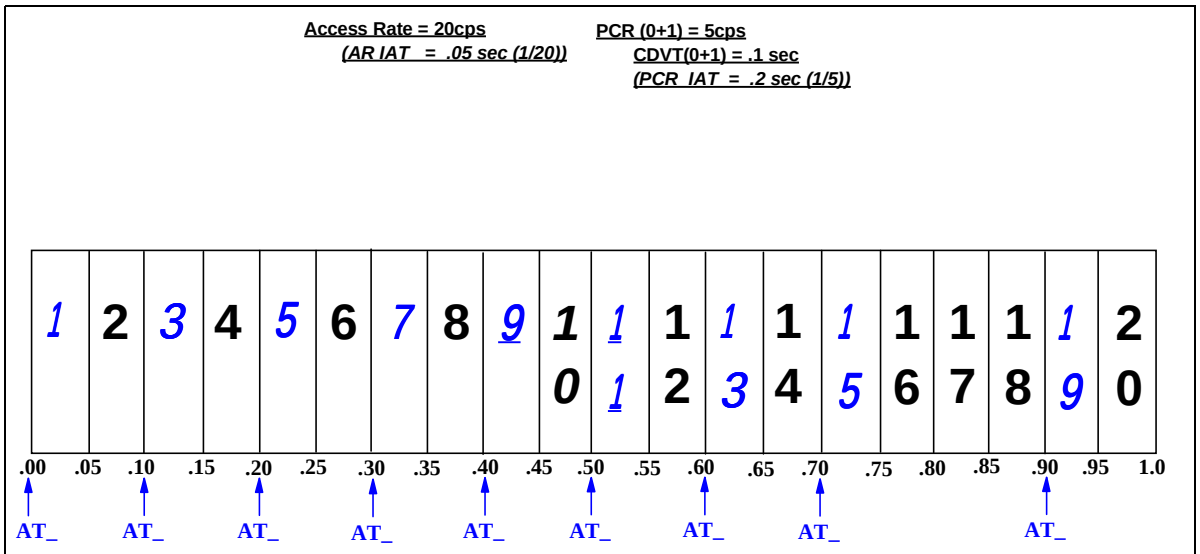

Figure 13

Hopefully you came up with the following:

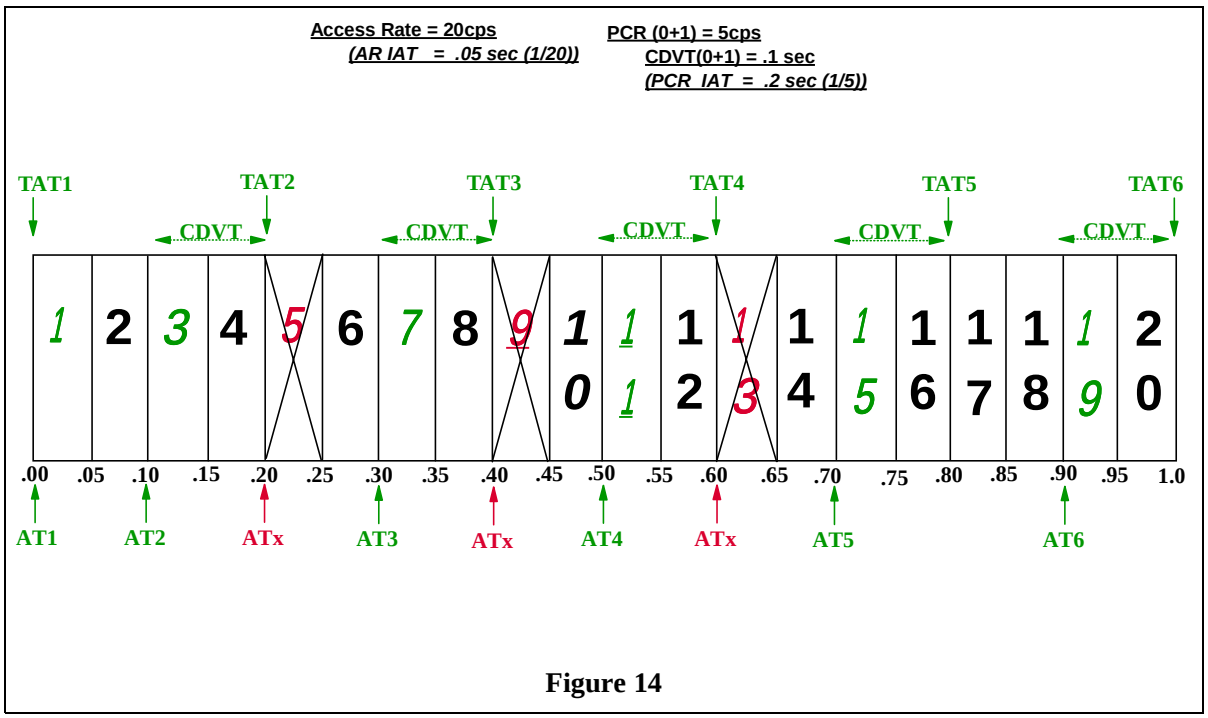

Figure 14

- Cell 1 is a CLP(0) data cell and is received at time .00sec. This is the first cell received so it has to be conforming.
  - *AT1* is set to .00sec.
  - TAT2 = .2sec.: AT1 was not later than TAT1 so TAT2 = TAT1 + PCR IAT.

- Cell 2 is an Idle Cell.
- Cell 3 is a CLP(0) data cell. Since it was not received before TAT2-CDVT it is conforming.
  - AT2 is set to .1sec.
  - TAT3 = .4sec.: AT2 was not later than TAT2 so TAT3 = TAT2 + PCR IAT.
- Cell 4 is an Idle Cell.
- Cell 5 is a CLP(0) data cell. Since it was received **before** TAT3-CDVT it is nonconforming and will be discarded.
  - AT is not reset.
  - TAT is not reset.
- Cell 6 is an Idle Cell.
- Cell 7 is a CLP(0) data cell. Since it was not received before TAT3-CDVT it is conforming.
  - AT3 is set to .3sec.
  - TAT4 = .6sec.: AT3 was not later than TAT3 so TAT4 = TAT3 + PCR IAT.
- Cell 8 is an Idle Cell.
- Cell 9 is a CLP(1) data cell. Since it was received **before** TAT4-CDVT it is nonconforming and will be discarded.
  - AT is not reset.
  - TAT is not reset.
- Cell 10 is an Idle Cell.
- Cell 11 is a CLP(1) data cell. Since it was not received before TAT4-CDVT it is conforming.
  - AT4 is set to .5sec.
  - TAT5 = .8sec.: AT4 was not later than TAT4 so TAT5 = TAT4 + PCR IAT.
- Cell 12 is an Idle Cell.
- Cell 13 is a CLP(0) data cell. Since it was received **before** TAT5-CDVT it is nonconforming and will be discarded.
  - AT is not reset.
  - TAT is not reset.
- Cell 14 is an Idle Cell.
- Cell 15 is a CLP(0) data cell. Since it was not received before TAT5-CDVT it is conforming.
  - AT5 is set to .7sec.
  - TAT6 = 1.0sec.: AT5 was not later than TAT5 so TAT6 = TAT5 + PCR IAT.
- Cells 16, 17, and 18 are Idle Cells.
- Cell 19 is a CLP(0) data cell. Since it was not received before TAT6-CDVT it is conforming.
  - AT6 is set to .9sec.
  - TAT7 = 1.2sec.: AT6 was not later than TAT6 so TAT7 = TAT6 + PCR IAT.
- Cell 20 is an Idle Cell.

So, the misbehaving user tried to send 9 cells and 6 were allowed. Let's look at some examples in which CDVT can be used to get around a common problem.

One Stage PCR/CDVT Policing: Example 8

Let's say that you're supporting a UBR type application with your PCR circuit. Applications that can use UBR are always variable bit rate type traffic generators which means that the data they send started out as frames (i.e. an IP Packet) which are then segmented into ATM cells. That frame worth of cells is supposed to be burst out at PCR but if *Shaping* is not implemented in the ATM source there is a good chance that many of the cells of a frame will really be sent out at the Access Rate (AR) with no regard to the PCR. The best way to demonstrate what this can do is to let you try determine the appropriate CDVT value for the following scenario. This is called the "learning to swim by being thrown into the pool method" of instruction.

Assume that the transmitting application produced frames that each took a maximum of 3 ATM cells to transport though the total number of cells sent in one second would still be 5. This means you could have conforming combinations like:

- One 3 cell frame and One 2 cell frame
- Two 2 cell frames and One 1 cell frame
- etc.

What CDVT value would you have to specify so that all the cells in the example on the next page would be conforming? Take as much time as you want to figure it out and please be prepared to explain how you determined your value. I'll go get some coffee and check back with you in 5 minutes.

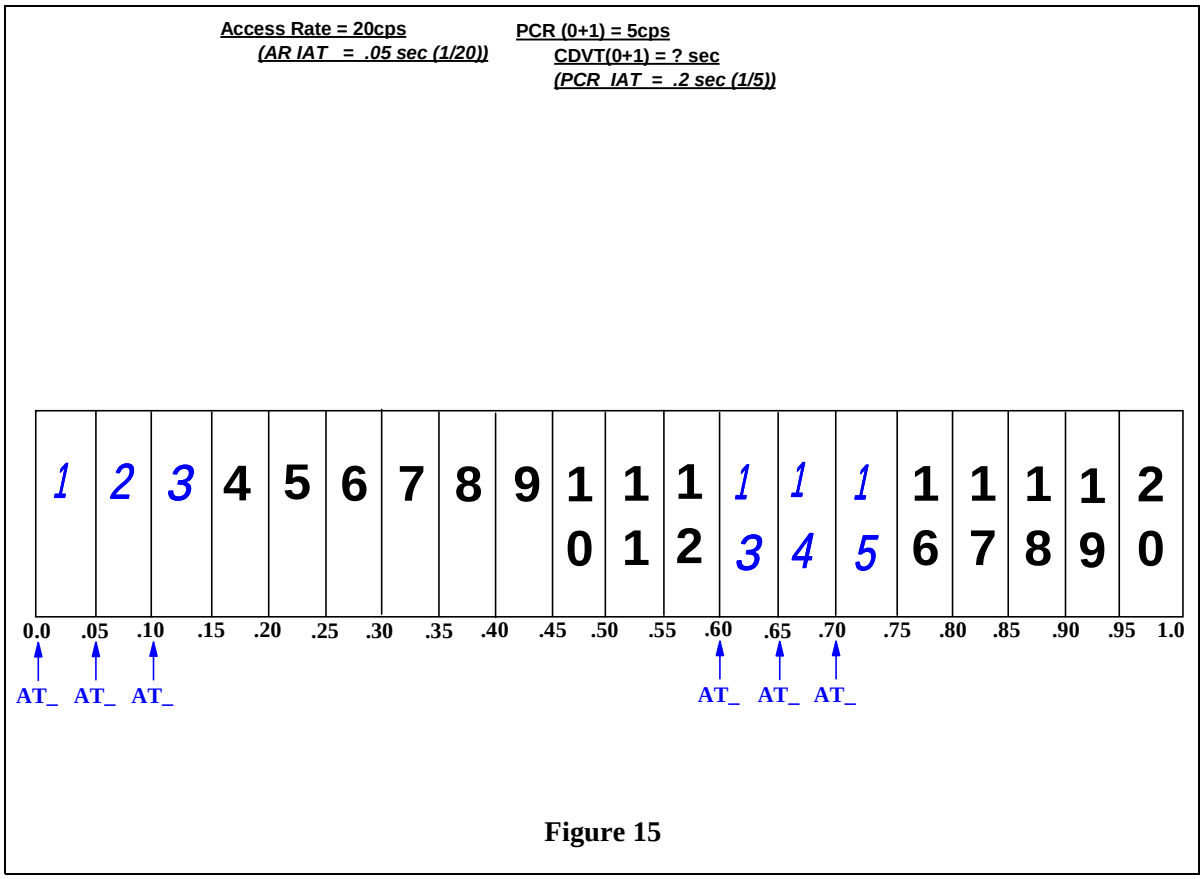

Figure 15

How did it go?

How many of you thought the CDVT should be .15sec? Let's look on the next page to see how that would work.

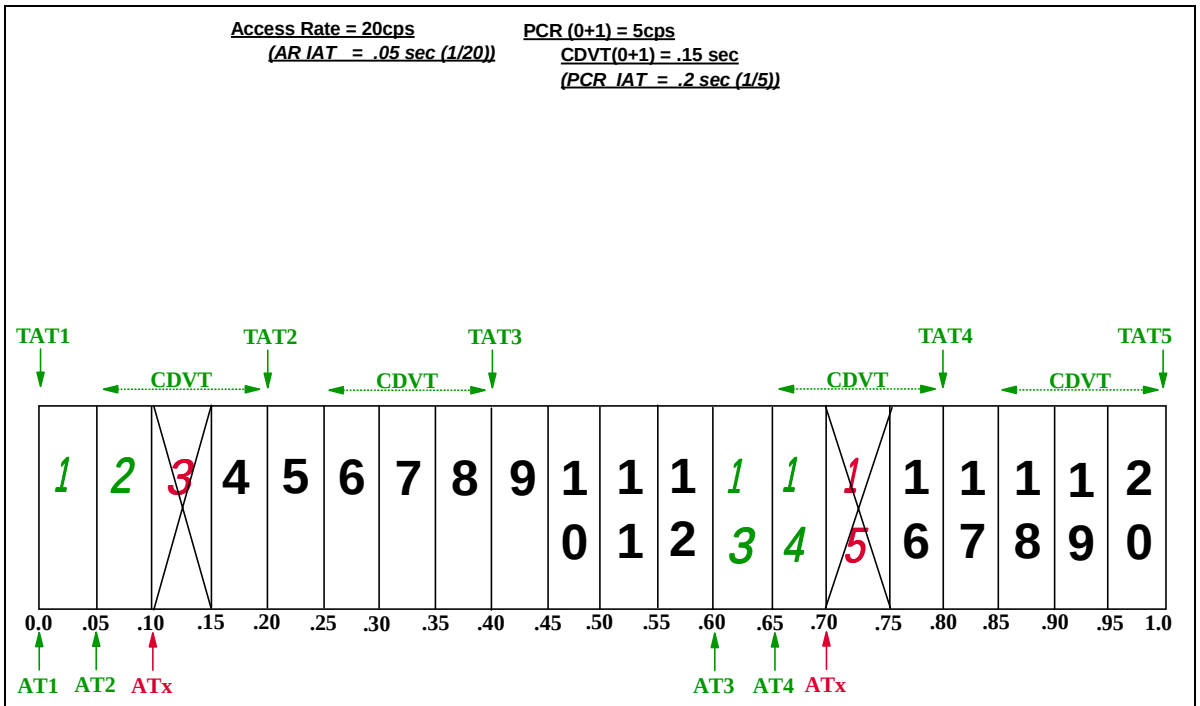

Figure 16

As shown, a CDVT of .15sec does **not** support 3 consecutive cells at the Access Rate. But why?

- Cell 1 is a CLP(0) data cell and is received at time .00sec. This is the first cell received so it has to be conforming.
  - *AT1* is set to .00sec.
  - TAT2 = .2sec.: AT1 was not later than TAT1 so TAT2 = TAT1 + PCR IAT.
- Cell 2 is a CLP(0) data cell. Since it was not received before TAT2-CDVT it is conforming.
  - AT2 is set to .05sec.
  - TAT3 = .4sec.: AT2 was not later than TAT2 so TAT3 = TAT2 + PCR IAT. **This is what causes the problem with a .15sec. CDVT. The expected arrival time of the next cell is TAT3 so even though Cell 3 is *under a CDVT line* that's TAT2's CDVT which was really applicable only to AT2. Remember, the *next TAT* is recalculated after the receipt of every conforming cell.**
- Cell 3 is a CLP(0) data cell. Since it was received **before TAT3-CDVT** it is nonconforming and will be discarded.
  - AT is not reset.
  - TAT is not reset.
- Cells 4 through 12 Idle Cells.
- Cell 13 is a CLP(0) data cell. Since it was not received before TAT2-CDVT it is conforming.

    - AT3 is set to .6sec.
    - TAT4 = .8sec.: AT3 was **later than TAT3** so TAT4 = AT3 + PCR IAT.
- Cell 14 is a CLP(0) data cell. Since it was not received before TAT4-CDVT it is conforming.
    - AT4 is set to .65sec.
    - TAT5 = 1.0sec.: AT4 was not later than TAT4 so TAT5 = TAT4 + PCR IAT.
- Cell 15 is a CLP(0) data cell. Since it was received **before TAT5-CDVT** it is nonconforming and will be discarded.
    - AT is not reset.
    - TAT is not reset.
- Cells 16 through 20 Idle Cells.

Thus a CDVT of .15sec really only supported two (not three) back to back cells at AR.

Here is a simple algorithm for determining the correct CDVT value you need to support a certain number of back to back cells.

1. Decide how big a contiguous burst you want to support.
2. Calculate earliest possible **AT** for the last cell of the burst.: *(Burst Size - 1) * AR IAT*. Sorry about the *-1* stuff but it doesn't work unless you do that. The ***earliest possible*** means you assume the first cell of the burst comes in at .00sec.. Trust me, the CDVT value you'll come up with will still work even if the first cell doesn't come in at .00sec..
3. Calculate earliest possible **TAT** for the **last cell** of the burst: *(Burst Size - 1) * PCR IAT*.
4. Subtract earliest possible TAT from earliest possible AT of last cell of the burst and you get the size of the CDVT you need.

To test it, plug in the numbers from the previous example:

1. Burst Size: 3
2. Earliest AT for last cell of burst: (3-1) * .05sec. = .1sec.
3. Earliest TAT for last cell of burst: (3-1) * .2sec. = .4sec.
4. CDVT to support burst of 3 cells: .4 - .1 = .3sec

Now let's see how a CDVT of .3sec. works.

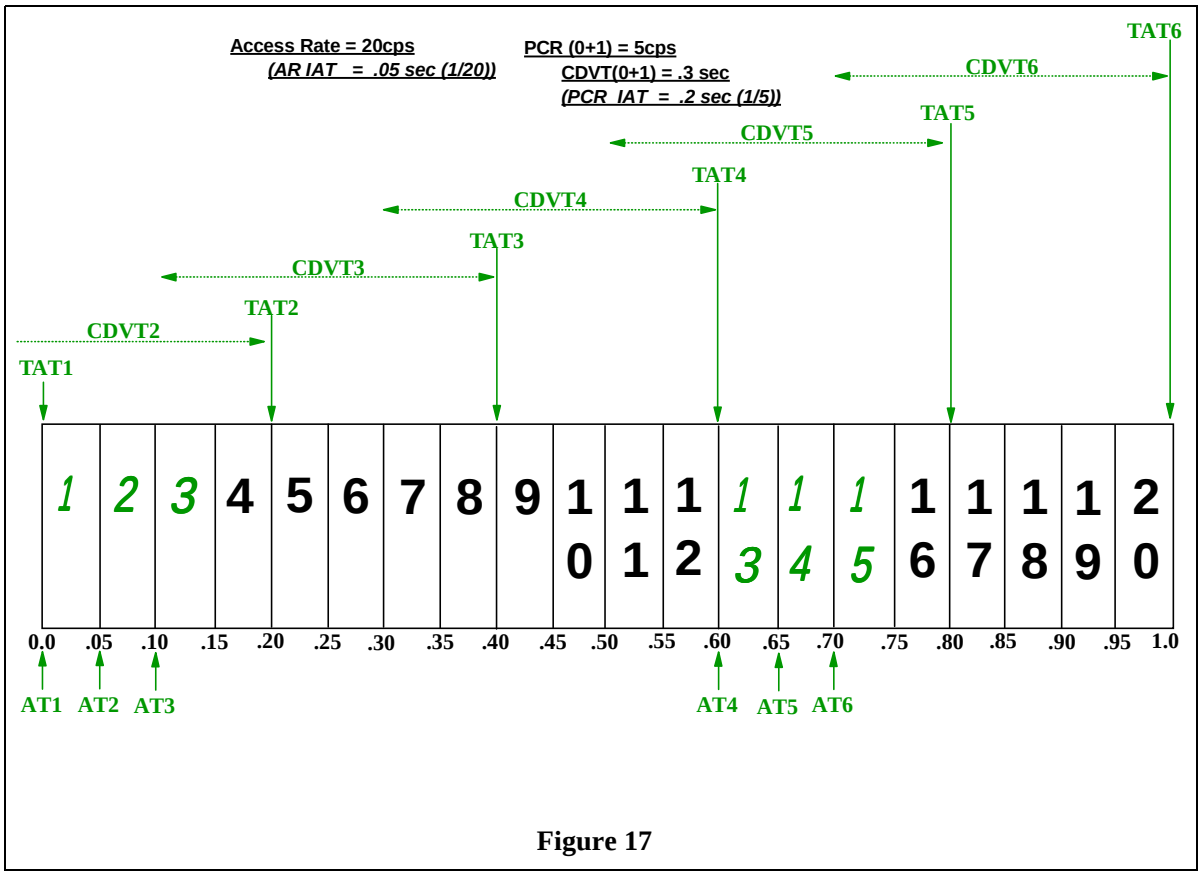

Figure 17

Notice how the CDVTs overlap.

- Cell 1 is a CLP(0) data cell and is received at time .00sec. This is the first cell received so it has to be conforming.
  - *AT1* is set to .00sec.
  - TAT2 = .2sec.: AT1 was not later than TAT1 so TAT2 = TAT1 + PCR IAT.
- Cell 2 is a CLP(0) data cell. Since it was not received before TAT2-CDVT it is conforming.
  - AT2 is set to .05sec.
  - TAT3 = .4sec.: AT2 was not later than TAT2 so TAT3 = TAT2 + PCR IAT.
- Cell 3 is a CLP(0) data cell. Since it was not received before TAT3-CDVT it is conforming. **Note that the process is exactly the same as in the previous example when this cell was discarded. The difference here is that the CDVT is now large enough to extend from TAT3 to Cell 3.**
  - AT3 is set to .1sec.
  - TAT4 = .6sec.: AT3 was not later than TAT3 so TAT4 = TAT3 + PCR IAT.
- Cells 4 through 12 Idle Cells.
- Cell 13 is a CLP(0) data cell. Since it was not received before TAT4-CDVT it is conforming.
  - AT4 is set to .6sec.
  - TAT5 = .8sec.: AT4 was **later than TAT4** so TAT5 = AT4 + PCR IAT.

- Cell 14 is a CLP(0) data cell. Since it was not received before TAT5-CDVT it is conforming.
  - AT5 is set to .65sec.
  - TAT6 = 1.0sec.: AT5 was not later than TAT5 so TAT6 = TAT5 + PCR IAT.
- Cell 15 is a CLP(0) data cell. Since it was not received before TAT6-CDVT it is conforming.
  - AT6 is set to .7sec.
  - TAT7 = 1.2sec.: AT6 was not later than TAT6 so TAT7 = TAT6 + PCR IAT. (off the chart)
- Cells 16 through 20 Idle Cells.

Six cells were conforming even though the PCR was only 5 but we've seen this sort of thing happen due to the CDVT value before.

For those of you who think making a large CDVT is a way of making "everything" conforming I've got bad news for you. The mathematicians who came up with all these algorithms designed them so that couldn't occur.

Let's see what happens when we slightly alter the arrival times of some of the data cells.

One Stage PCR/CDVT Policing: Example 9

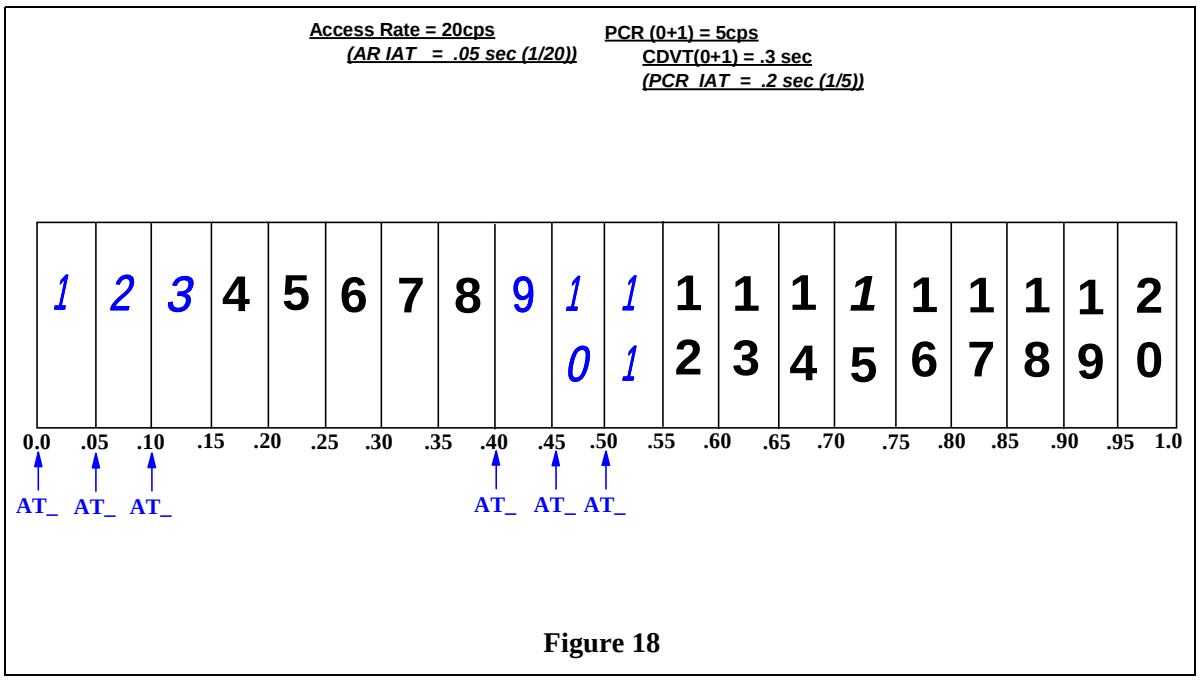

Figure 18

All I did is move up the second 3 cell burst a bit. Will all the data cells still be conforming?

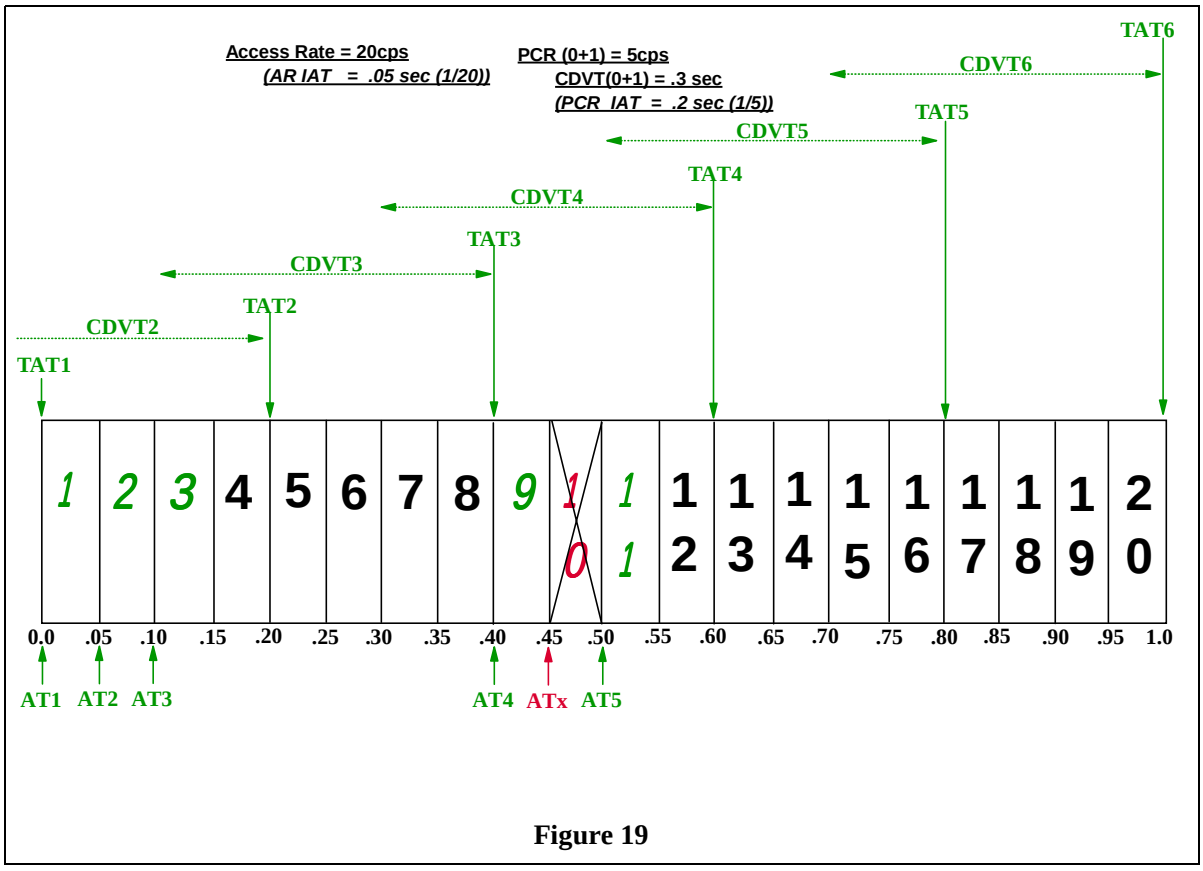

Figure 19

- Cell 1 is a CLP(0) data cell and is received at time .00sec. This is the first cell received so it has to be conforming.
  - *AT1* is set to .00sec.
  - TAT2 = .2sec.: AT1 was not later than TAT1 so TAT2 = TAT1 + PCR IAT.
- Cell 2 is a CLP(0) data cell. Since it was not received before TAT2-CDVT it is conforming.
  - AT2 is set to .05sec.
  - TAT3 = .4sec.: AT2 was not later than TAT2 so TAT3 = TAT2 + PCR IAT.
- Cell 3 is a CLP(0) data cell. Since it was not received before TAT3-CDVT it is conforming.
  - AT3 is set to .1sec.
  - TAT4 = .6sec.: AT3 was not later than TAT3 so TAT4 = TAT3 + PCR IAT.
- Cells 4 through 8 Idle Cells.
- Cell 9 is a CLP(0) data cell. Since it was not received before TAT4-CDVT it is conforming.
  - AT4 is set to .4sec.
  - TAT5 = .8sec.: AT4 was **not later than TAT4** so TAT5 = TAT4 + PCR IAT. **In the previous example when the fourth data cell came in it was AFTER TAT4 so AT4 was used to calculate the next TAT.**
- Cell 10 is a CLP(0) data cell. Since it was received **before TAT5-CDVT** it is nonconforming and will be discarded. Remember, the next conforming data cell's TAT is

TAT5. If you follow the line CDVT for TAT5 (or just subtract the CDVT value from TAT5) you see the earliest a conforming cell could arrive is .5sec..
    - AT is not reset.
    - TAT is not reset.
- Cell 11 is a CLP(0) data cell. Since it was not received before TAT5-CDVT it is conforming.
    - AT5 is set to .5sec.
    - TAT6 = 1.0sec.: AT5 was not later than TAT5 so TAT6 = TAT5 + PCR IAT.
- Cells 12 through 20 Idle Cells.

Why the difference? The basic rule (for non-mathematicians) is that if you want to repeat the maximum conforming burst you have to wait until the actual TAT (not TAT-CDVT) of the last conforming cell of that burst *to allow the GCRA (Generic Cell Rate Algorithm) to become zero.*

If you read the *Preface* (Shame on you if you didn't. Go back right now and read it!) you may remember I solved my first experience with massive cell discards by using a *big* CDVT value. I found out later that the problem had been that although the LAN switches had been configured with reasonable PCRs, for UBR traffic, they just ignored the configured value and always transmitted as if the PCR had been set to the media speed (100mbps). Since *Shaping* wasn't available at the time is should be clear now why I was having a problem and why the larger CDVT value helped.

Unfortunately, as mentioned earlier, CDVT is a network not a user specified circuit parameter. Most service providers will not let you specify the CDVT **they** configure for your circuit. Since, as we've seen, the CDVT value can allow a few more cells to be conforming than implied by the PCR this is understandable. Thus, in most situations you'll have to address nonconformance problems by increasing the PCR and/or implementing *Shaping* on the traffic generating device.

That's probably more than enough on CDVT.

Now that you hopefully have a good understanding of the basics of PCR and CDVT the rest should be easy. Let's look at *Two Stage* PCR/CDVT Policing.

Two Stage PCR/CDVT Policing - No Tagging

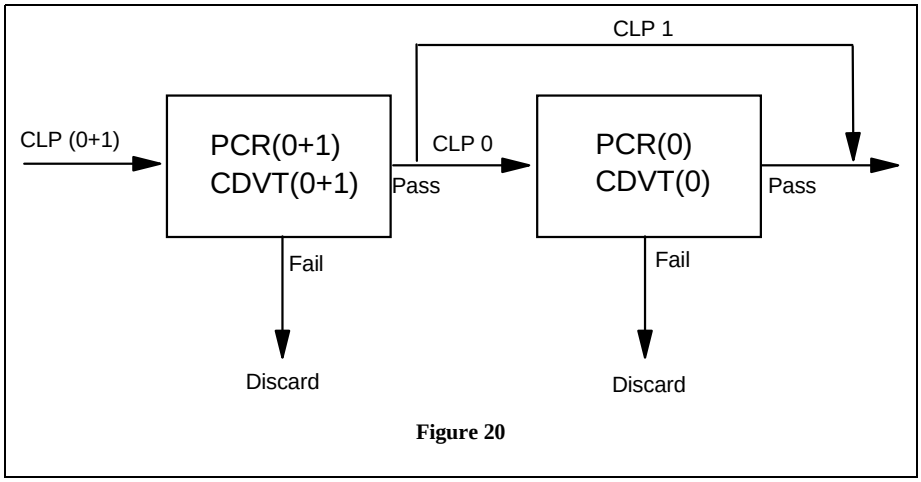

Figure 20

Figure 20 shows the first *Two Stage* PCR/CDVT algorithm. Notice that CLP(1) cells that make it through the first test are not subjected to the second. On the surface this looks unfair. Why should the *bad* cells that had been marked nonconforming elsewhere in the network get an automatic pass while the *good* conforming cells be subjected to an additional test? It's like the police at a roadblock waving really nasty looking people through but doing full car searches of all the sweet old ladies!

The basic reason for this is that there is no QOS guarantee for nonconforming cells. The first PCR test is mainly used to protect the network against congestion, the second is used to police circuits which have a guaranteed QOS. The best example of where this could be used would be a VPC circuit.

The switch policing a VPC has only one set of traffic descriptors for the circuit even though it may be tunneling multiple VCCs. Let's assume a VPC is carrying two VCCs with one passing UBR traffic and the other (reserved) VBR traffic. As mentioned earlier many end devices that generate UBR traffic create the ATM cells with CLP=1. To protect against congestion the first stage test makes sure that the total VPC usage doesn't exceed the agreed on rate. This test appears in **all** the different policing algorithms. The second test ensures that the QOS guaranteed traffic (the reserved VBR) is conforming. Since that traffic is a subset of the total VPC traffic, the second stage's PCR and CDVT values must be different than the first stage's.

Let's look at an example.

Two Stage PCR/CDVT (No Tagging) Policing: Example 1

Determine which cells are conforming and which are nonconforming in the scenario below. Note that the PCR and CDVT are different for each stage.

Use the following process to determine conformance:

1. Using a pencil, mark up the initial diagram with ATs, TATs, and CDVT lines to determine which cells pass the first PCR/CDVT test. Make sure you put an X through the cells that would be discarded.
2. Erase the ATs, TATs, and CDVT lines from the first pass but keep the X'd out cells marked. Use this as input to the second test filling in the new ATs, TATs, and CDVTs using the second stage's PCR and CDVT values. Remember you can ignore any of the cells that have CLP=1 or had been discarded by the first stage (almost like they were Idle Cells) since the second test is only concerned with the conforming cells that have CLP=0.

Do the first stage's policing and then check to see if you got it right before doing the second stage.

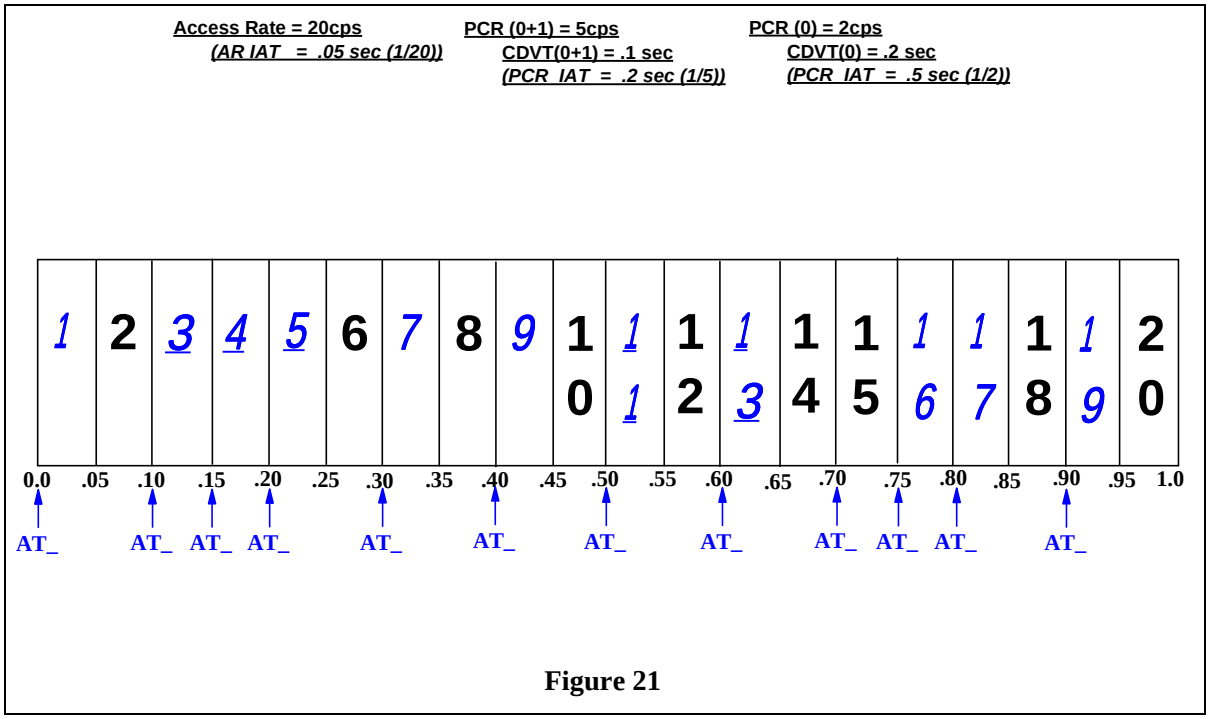

Figure 21

The results of the first stage policing are on the next page.

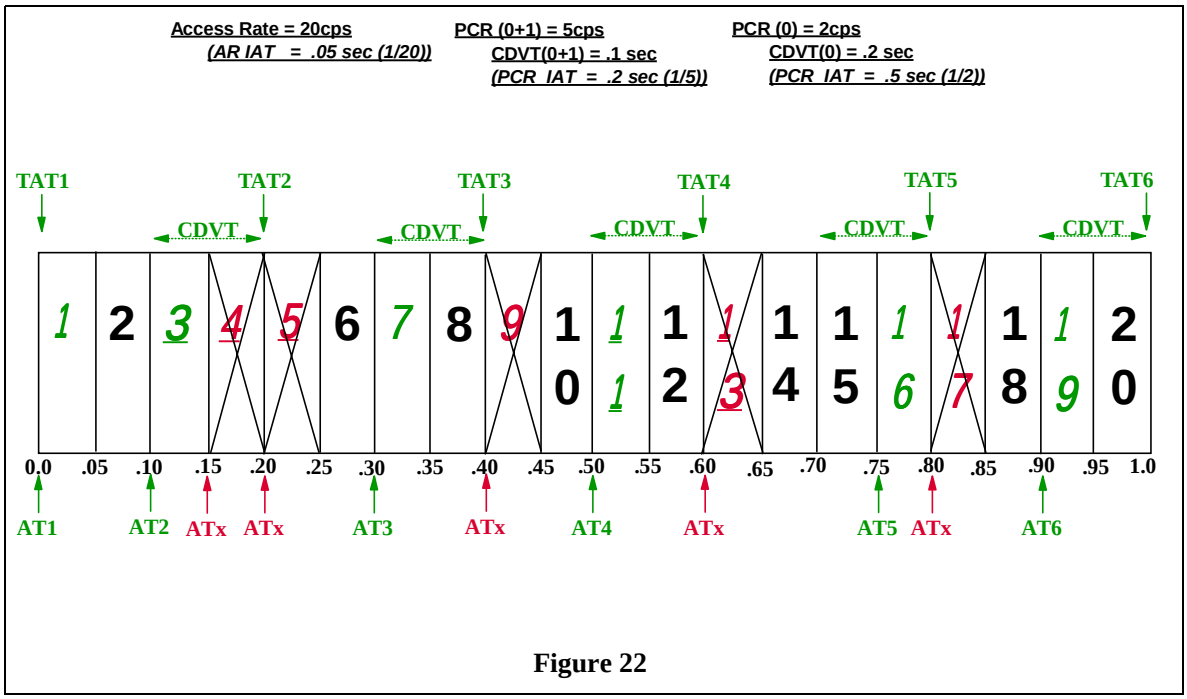

Figure 22

- Cell 1 is a CLP(0) data cell and is received at time .00sec. This is the first cell received so it has to be conforming.
  - *AT1* is set to .00sec.
  - TAT2 = .2sec.: AT1 was not later than TAT1 so TAT2 = TAT1 + PCR IAT.
- Cell 2 is an Idle Cell.
- Cell 3 is a CLP(1) data cell. Since it was not received before TAT2-CDVT it is conforming.
  - AT2 is set to .1sec.
  - TAT3 = .4sec.: AT2 was not later than TAT2 so TAT3 = TAT2 + PCR IAT.
- Cell 4 is a CLP(1) data cell. Since it was received **before** TAT3-CDVT it is nonconforming and will be discarded.
  - AT is not reset.
  - AT is not reset.
- Cell 5 is a CLP(1) data cell. Since it was received **before** TAT3-CDVT it is nonconforming and will be discarded.
  - AT is not reset.
  - TAT is not reset.
- Cell 6 is an Idle Cell.
- Cell 7 is a CLP(0) data cell. Since it was not received before TAT3-CDVT it is conforming.
  - AT3 is set to .3sec.
  - TAT4 = .6sec.: AT3 was not later than TAT3 so TAT4 = TAT3 + PCR IAT.
- Cell 8 is an Idle Cell.
- Cell 9 is a CLP(0) data cell. Since it was received **before** TAT4-CDVT it is nonconforming and will be discarded.

    - AT is not reset.
    - TAT is not reset.
- Cell 10 is an Idle Cell.
- Cell 11 is a CLP(1) data cell. Since it was not received before TAT4-CDVT it is conforming.
    - AT4 is set to .5sec.
    - TAT5 = .8sec.: AT4 was not later than TAT4 so TAT5 = TAT4 + PCR IAT.
- Cell 12 is an Idle Cell.
- Cell 13 is a CLP(1) data cell. Since it was received **before** TAT5-CDVT it is nonconforming and will be discarded.
    - AT is not reset.
    - TAT is not reset.
- Cells 14 and 15 are Idle Cells.
- Cell 16 is a CLP(0) data cell. Since it was not received before TAT5-CDVT it is conforming.
    - AT5 is set to .75sec.
    - TAT6 = 1.0sec.: AT5 was not later than TAT5 so TAT6 = TAT5 + PCR IAT.
- Cell 17 is a CLP(0) data cell. Since it was received **before** TAT6-CDVT it is nonconforming and will be discarded.
    - AT is not reset.
    - TAT is not reset.
- Cell 18 is an Idle Cell.
- Cell 19 is a CLP(0) data cell. Since it was not received before TAT6-CDVT it is conforming.
    - AT6 is set to .9sec.
    - TAT7 = 1.2sec.: AT6 was not later than TAT6 so TAT7 = TAT6 + PCR IAT.
- Cell 20 is an Idle Cell.

Now go ahead and try the second stage policing. Remember the PCR and the CDVT values are different than in the first stage. Also, only the data cells that passed the first stage **and** are CLP=0 are tested. The cell slots containing Idle, Discarded, and CLP=1 cells are only used as place holders. Try to use a different technique to identify cells discarded by the second stage so it's easier to differentiate them from the ones discarded by the first stage. In my examples, I still color those cell number red but use dotted lines to X out the cells discarded by the second stage.

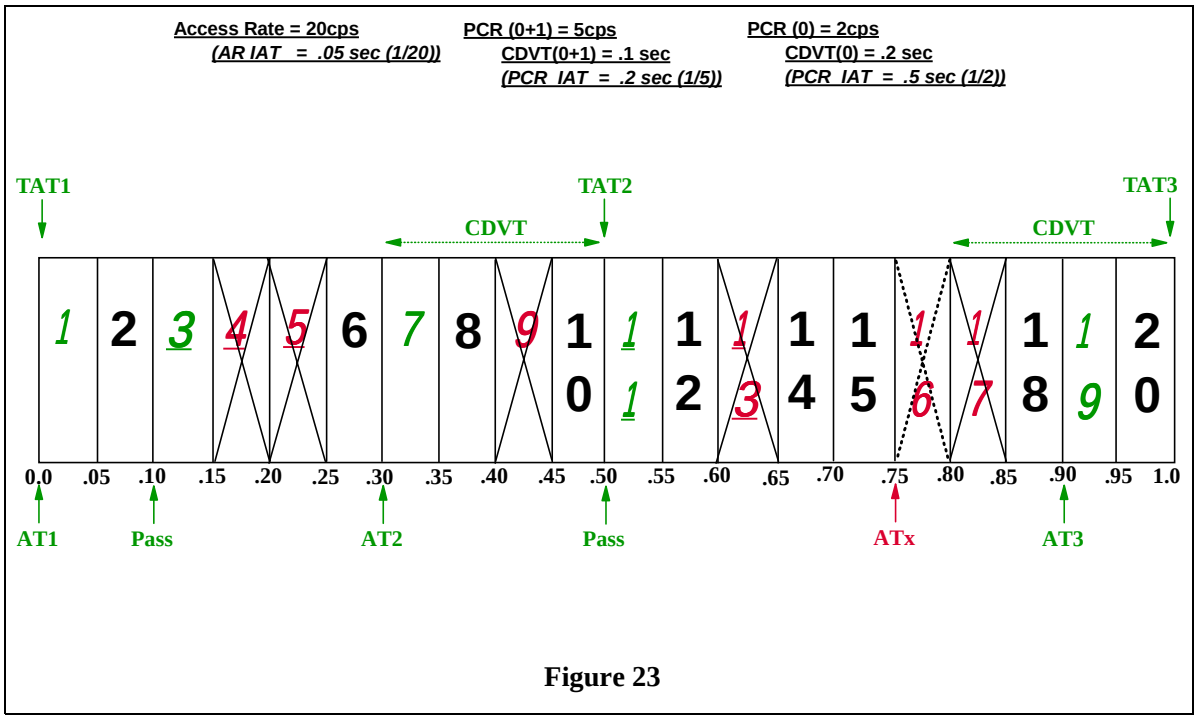

Figure 23

- Cell 1 is a CLP(0) data cell and is received at time .00sec. This is the first cell received so it has to be conforming.
  - *AT1* is set to .00sec.
  - TAT2 = .5sec.: AT1 was not later than TAT1 so TAT2 = TAT1 + PCR IAT.
- Cell 2 is an Idle Cell.
- Cell 3 is a conforming CLP(1) data cell. **Remember, in the Second Stage CLP(1) cells, like Idle Cells, are only used as place holders.**
- Cells 4 and 5 had been discarded by the First Stage.
- Cell 6 is an Idle Cell.
- Cell 7 is a CLP(0) data cell. Since it was not received before TAT2-CDVT it is conforming.
  - AT2 is set to .3sec.
  - TAT3 = 1.0sec.: AT2 was not later than TAT2 so TAT3 = TAT2 + PCR IAT.
- Cell 8 is an Idle Cell.
- Cell 9 had been discarded by the First Stage.
- Cell 10 is an Idle Cell.
- Cell 11 is a conforming CLP(1) data cell.
- Cell 12 is an Idle Cell.
- Cell 13 had been discarded by the First Stage.
- Cells 14 and 15 are Idle Cells.
- Cell 16 is a CLP(0) data cell. Since it was received **before** TAT3-CDVT it is nonconforming and will be discarded. **So here is a cell discarded by the Second Stage.**
  - AT is not reset.
  - TAT is not reset.
- Cell 17 had been discarded by the First Stage.

- Cell 18 is an Idle Cell.
- Cell 19 is a CLP(0) data cell. Since it was not received before TAT3-CDVT it is conforming.
  - AT3 is set to .9sec.
  - TAT4 = 1.5sec.: AT3 was not later than TAT3 so TAT4 = TAT5 + PCR IAT (off the chart).
- Cell 20 is an Idle Cell.

Due to the first stage's CDVT, 6 CLP(0) and CLP(1) data cells got through although the PCR was only 5 cps. Due to the second stage's CDVT, 3 CLP(0) data cells got through even though the PCR was only 2cps.

Since all this is so much like the *Single Policing* process for which many exercises have already been done, this one example for *Two Stage* without tagging should be sufficient. Now let's look at the third and final PCR/CDVT policing algorithm.

Two Stage PCR/CDVT Policing - Tagging

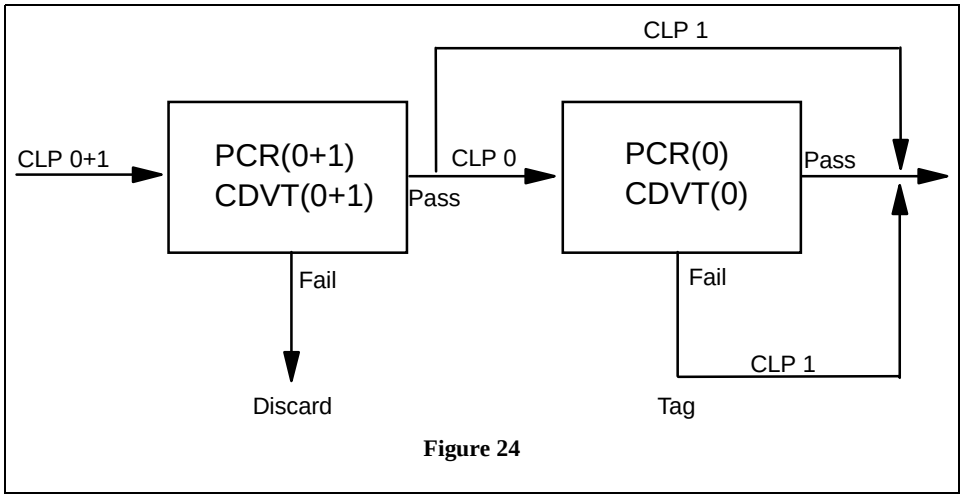

Figure 24

As you can see the only difference between this algorithm and the previous is that the second stage *tags* nonconforming cells (sets CLP=1) instead of discarding them. Let's go right to an example.

Two Stage PCR/CDVT (Tagging) Policing: Example 1

All the PCR and CDVT values are the same as the previous example. The data cell (tagged and un-tagged) sequence is also the same.

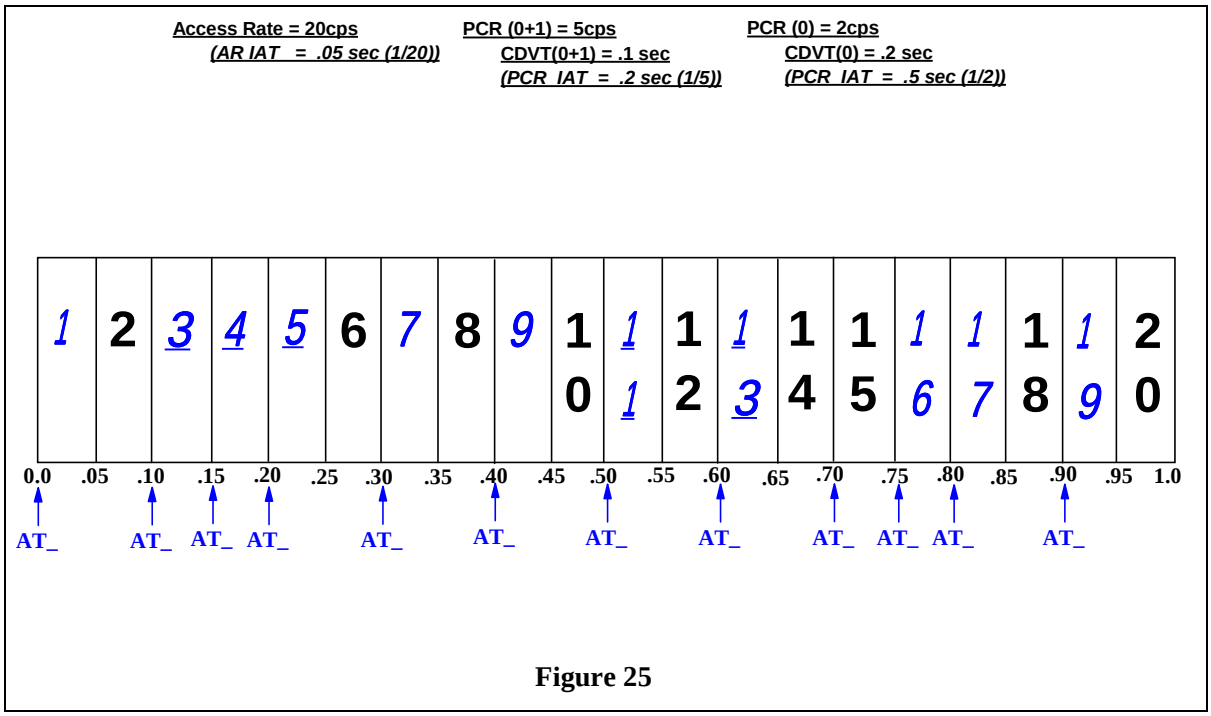

Figure 25

The results of the first PCR/CDVT test are the same as in the last example as you should have expected.

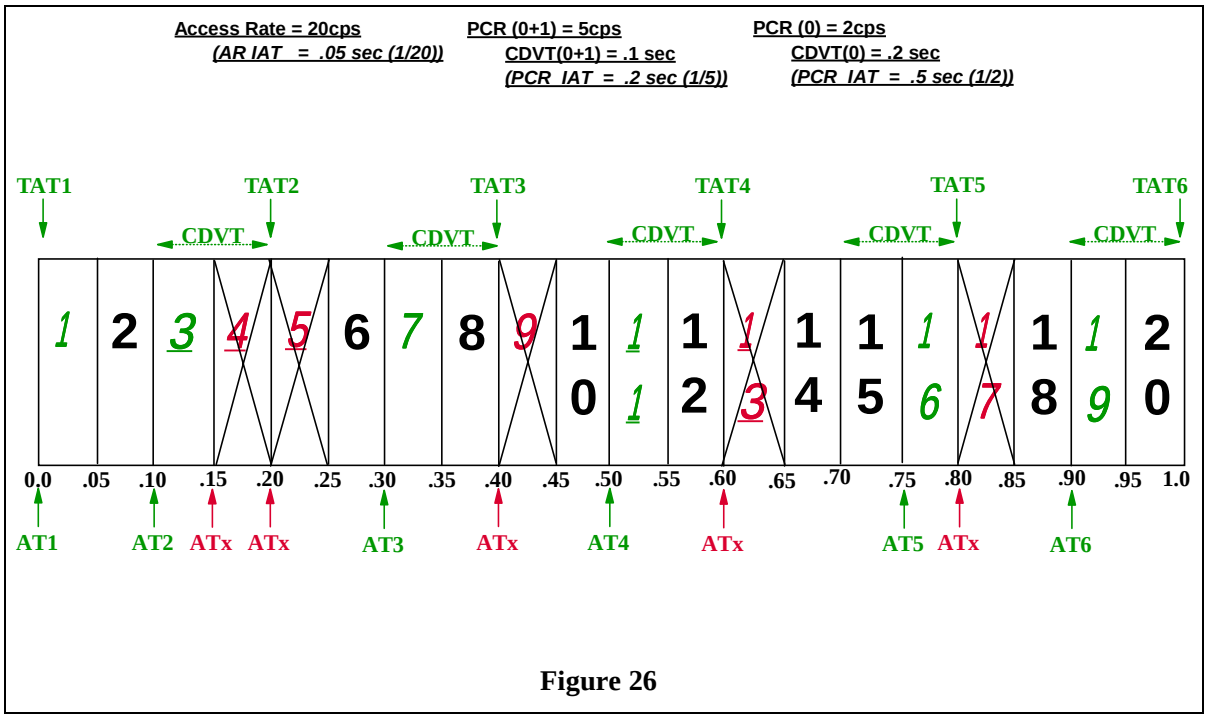

Figure 26

Now let's see how the second PCR/CDVT testing produces different results. Without even doing the analysis, what do you think will be the difference?

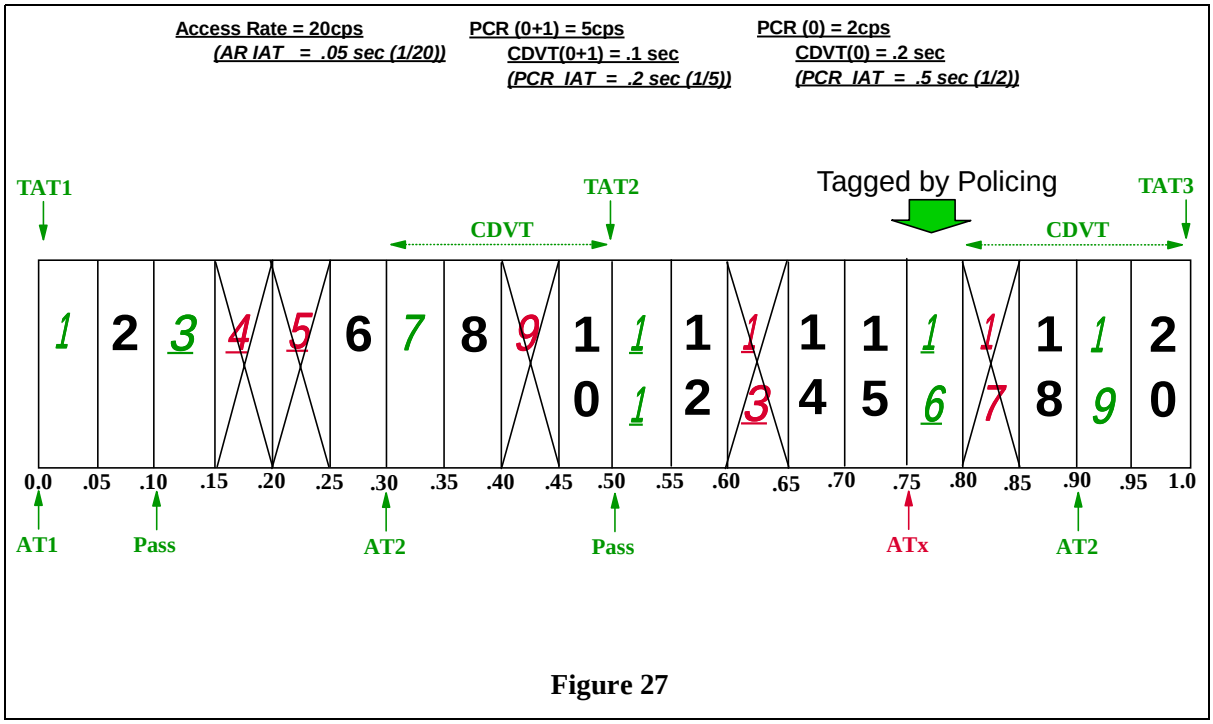

Figure 27

As most of you had probably realized, the **ONLY** difference from the non-tagging example is that Cell 16 is tagged and passed through instead of being discarded. Since everything else is the same, I'll only provide the cell by cell explanation for Cell 16.

- Cell 16 is a CLP(0) data cell. Since it was received before TAT3-CDVT it is nonconforming. It will be tagged (CLP set to 1) and passed into the network. **Note that AT and TAT are not reset since the cell was nonconforming.**
  - AT is **not** reset.
  - TAT is **not** reset

Hopefully, by now you are comfortable with how *One Stage* and *Two Stage* PCR/CDVT policing operates. This will make understanding the policing options for Variable Bit Rate traffic (VBR) much easier.

VBR Traffic Policing

Variable Bit Rate (VBR) describes traffic which can be characterized as *on/off* or, more commonly, as being *bursty*. The degree of traffic's *burstiness* can be determined by dividing its peak rate by its average rate. The larger the result, the *burstier* the traffic. Thus, traffic with a peak rate of 20cps and an average rate of 10cps has a *burstiness* of 2 (20/10) and is therefore less *bursty* than traffic with a peak rate of 20cps and an average rate of 5cps (20/5=4). TCP/IP is a good example of VBR traffic since a window of frames will be sent (the *on* period) followed by a wait (the *off* period) for an acknowledgment from the other end.

There's really no point in allocating network bandwidth for the *off* periods. By reserving bandwidth which is closer to the average than the peak you can *fit* more connections into the finite amount of bandwidth available in the network. This allows you to achieve *statistical gain* which is a fancy term for rational bandwidth overbooking. The hope is that when a Circuit A enters its *on* state enough other circuits will be in their *off* states so that there will be enough bandwidth in the network to carry Circuit A's traffic. To improve the likelihood of this actually occurring a special class of traffic descriptors have been designed to police VBR circuits to ensure they have the degree of *burstiness* which was promised when the network accepted the connection.

There are three policing algorithms involving the specification of an average rate:

- PCR/CDVT (0+1), SCR/MBS (0+1) - No Tagging
- PCR/CDVT (0+1), SCR/MBS (0) - No Tagging
- PCR/CDVT (0+1), SCR/MBS (0) - Tagging

Note each is *a Two Stage* algorithm.

PCR/CDVT (0+1), SCR/MBS (0+1) - No Tagging

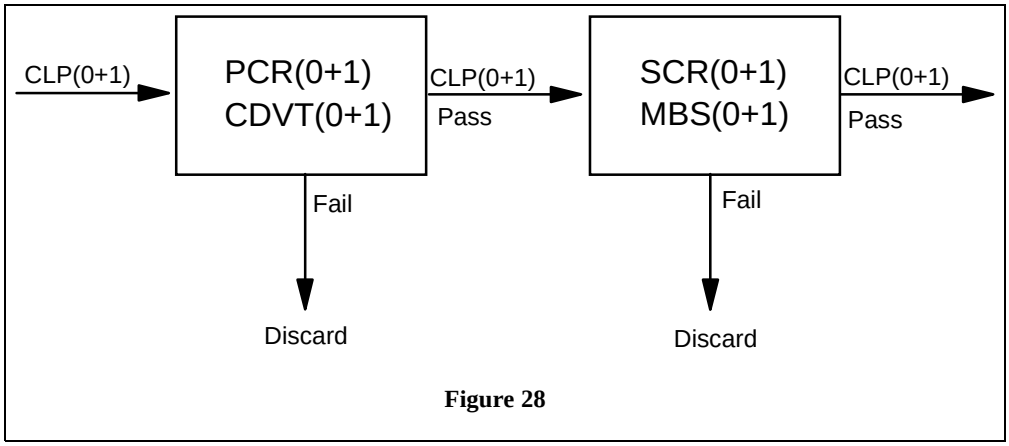

Figure 28

In addition to our old friends, PCR and CDVT, there are two new parameters..

- **Sustainable Cell Rate (*SCR*)**: This is the average rate, in cells per second, at which a virtual circuit can transmit. The specified value must be less than or equal to the PCR.
- **Maximum Burst Size (*MBS*)**: This is the maximum number of cells the connection can send back to back at Peak Cell Rate. The specified value must be less than or equal to the total number of cells per second allowed by the SCR.

Note that **both** the PCR/CDVT and the SCR/MBS tests are applied to all cells regardless of the state of their CLP bits (0+1). The second stages of the *Two Stage* PCR/CDVT algorithms we covered in the last section only operated on CLP(0) cells.

With regard to conformance validation:

- **First Stage**: The connection can't send at a rate greater than allowed by PCR(0+1)/CDVT(0+1).
- **Second Stage**: There are two validations.
  - *Burst Conformance*
    - The connection can only send back to back cells at the rate conforming to PCR(0+1)/CDVT(0+1) in bursts (groups) of cells no larger than MBS; smaller is ok. Now, how would **you** implement this stage? Just keep a counter to see if a burst is too long, right? Wrong! The value that's specified for MBS actually gets transformed into a time value called *Burst Tolerance* (BT) which is calculated via the equation: *BT = ((MBS -1) * (1/SCR + 1/PCR)) + CDVT*. What you're supposed to do is add this number to the arrival time of the first cell of a burst. The result will point to the **last** time a burst compliant cell can be received. If a data cell comes in at the first PCR TAT **after** this time the burst would be too long and that cell (not the entire burst) would be considered naughty. The CDVT

(same value used in the *First Stage*) is added on to allow for longer bursts that can occur due to *cell clumping*.

- Since a burst can be shorter than MBS how do you tell when one burst ends and the next one starts? Simple, there has to be *at least* one PCR TAT between the end of one burst and the start of the other which does not a contain a data cell that is conforming to PCR/CDVT (ie. an Idle Cell). In reality, this *gap* is usually larger than just one PCR TAT, especially when bursts of the maximum size are being sent (*to allow the GCRA to become zero).*

- *SCR Conformance*: The connection can't send a total of more cells in one second than specified by the SCR value. In other words, if you add up all the *conforming* cells in each burst, the total shouldn't exceed SCR.

This may all be confusing right now but some examples will help.

SCR(0+1)/MBS(0+1) Policing: Example 1

To simplify things, CDVT has been set to 0. Don't try this one yourself as there are few new conventions I'll be explaining.

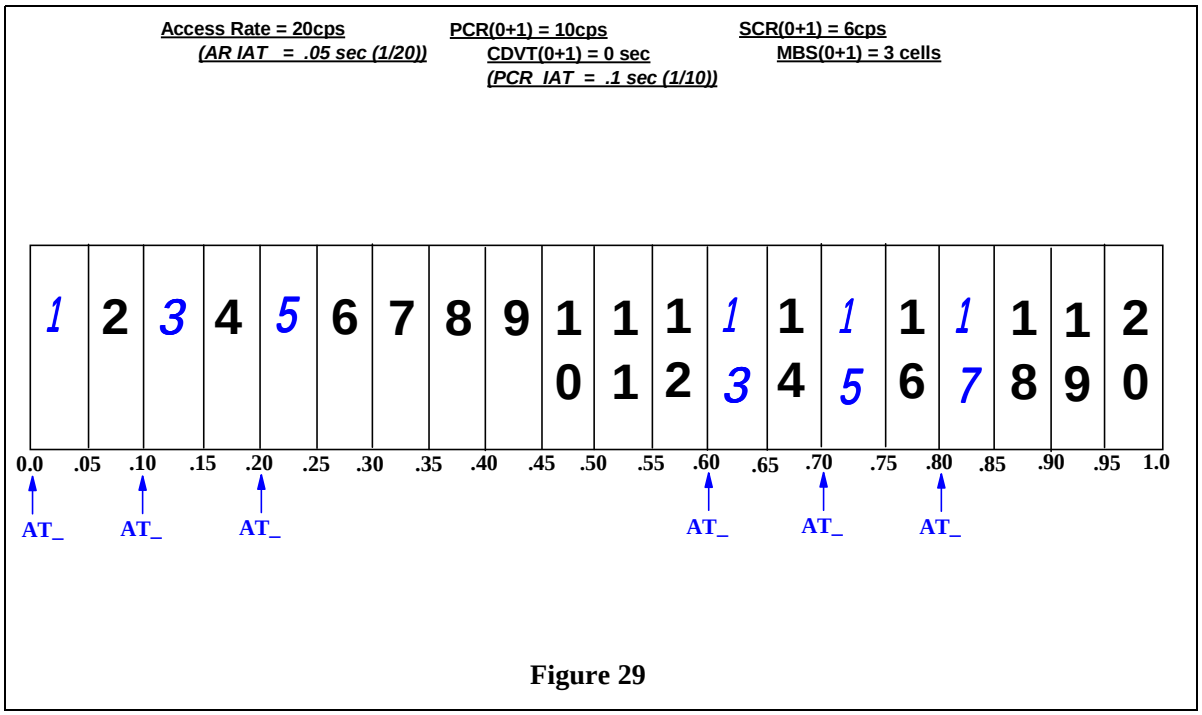

Figure 29

First, you have to perform exactly the same PCR/CDVT policing we did before. This will result in the following.

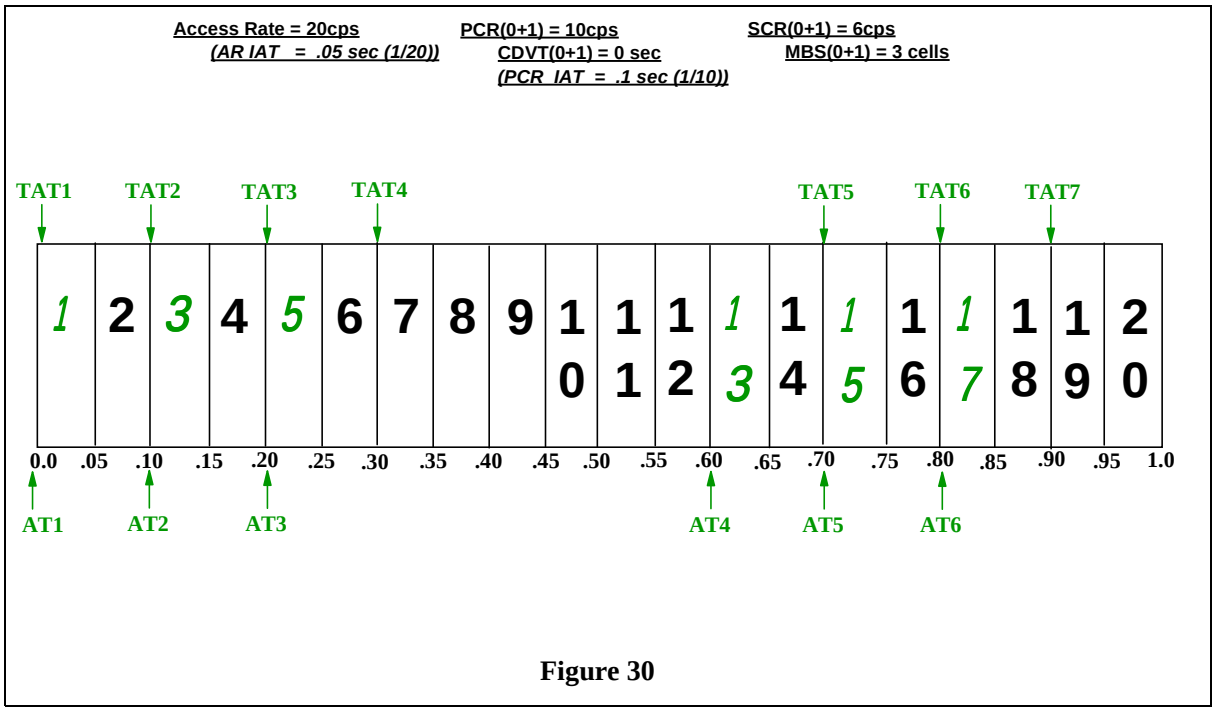

Figure 30

All the data cells were conforming. Now for the second stage all you have to do is make sure the MBS and SCR are not exceeded. To make the basic process easier to understand we'll take the following liberties:

- Instead of calculating *Burst Tolerance* we'll just keep a Burst Counter. SCR compliance will be tracked via a Cell Counter.
- We'll assume a *gap* between bursts of at least one PCR TAT will be sufficient no matter the size of the burst or the CDVT value.

Now, using the above figure as input, the SCR/MBS policing should result in the following:

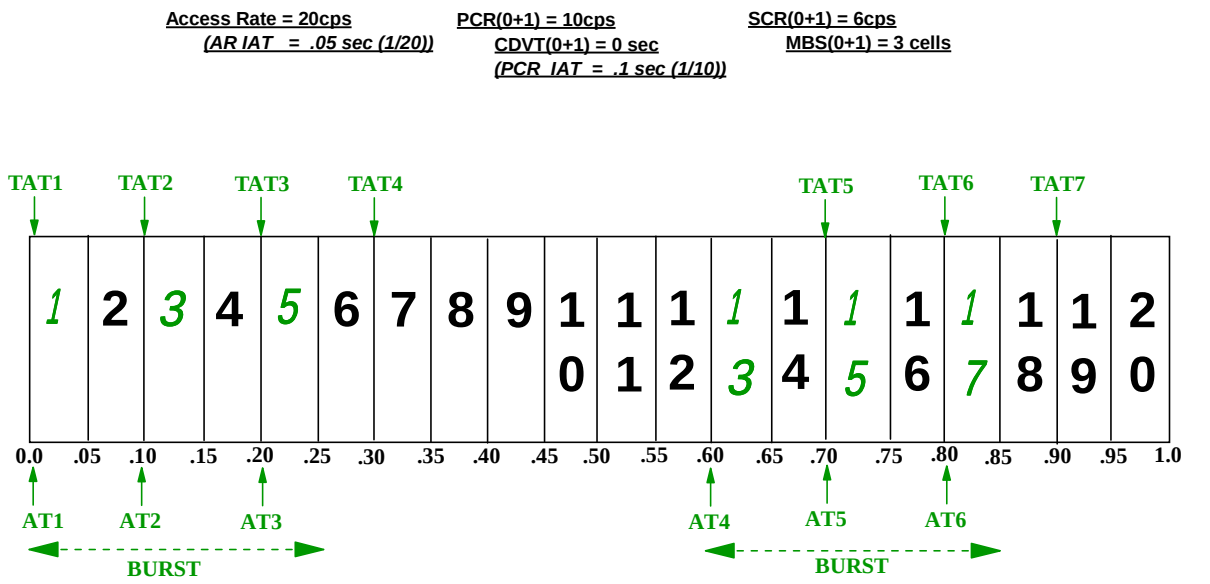

Figure 31

- Cell 1 is a CLP(0) data cell and was received at time .00sec. It is the first data cell as well as the first cell of a burst so:
  - **Burst Count = 1**: This is less than or equal to the MBS (3) so it is Burst conforming.
  - **Cell Count = 1**: This is less than or equal to the SCR cps (6) so it is SCR conforming.
- Cell 2 is an Idle Cell.
- Cell 3 is a CLP(0) data cell.
  - **Burst Count = 2**: Since the cell was received at **or before** the TAT from the previous data cell it's considered part of the same burst. Since the result of incrementing the burst counter would be less than or equal to the MBS (3) it is Burst conforming so the counter is incremented.
    - First question: Why is the ***"or before"*** in the calculation? Remember, the CDVT used in the first stage could have allowed a cell that came in earlier than the TAT.
    - Second question: Why don't we have to worry about the cell having come in earlier than the CDVT allowance? It's because that kind of cell would have been discarded by the first stage so it's safe to consider any cell that comes in earlier than the expected TAT is conforming.
  - **Cell Count = 2**: Since the cell was Burst conforming we now see if the result of incrementing the Cell Count would be less than or equal to the SCR cps (6). It is, so the cell is conforming to the SCR and we increment the counter.
- Cell 4 is an Idle Cell.
- Cell 5 is a CLP(0) data cell.
  - **Burst Count = 3**: Since the cell was received at or before the TAT from the previous data cell it's considered part of the same burst. Since the result of incrementing the burst counter would be less than or equal to the MBS (3) it is Burst conforming so the counter is incremented. Note that we are now at the maximum buffer size (3). If we

get another data cell at the next PCR TAT (TAT4) the burst would be too long and that cell would be discarded.
    - **Cell Count = 3**: Since the cell was Burst conforming we now see if the result of incrementing the Cell Count would be less than or equal to the SCR cps (6). It is so the cell is conforming to the SCR and we increment the counter.
- Cells 6 is an Idle Cell.
- Cell 7 is an Idle Cell **BUT** it's where the next data cell at PCR rate would have been (TAT4). This means the burst containing Cells 1, 3, and 5, is over. This is the *gap* I talked about earlier. What does this do to the counters?
    - **Burst Count = 0**: We reset the Burst Counter to prepare for the next burst.
    - **Cell Count = 3**: We **don't** reset the Cell Counter since that keeps track of the total number of cells we can send in one second.
- Cells 8 through 12 are Idle Cells.
- Cell 13 is a CLP(0) data cell.
    - **Burst Count = 1**: Since there was a PCR TAT *gap* after the last conforming data cell (TAT4) this is the first cell of a new burst. The count is less than or equal to the MBS (3) so it is Burst conforming.
    - **Cell Count = 4**: Since the cell was Burst conforming we now see if the result of incrementing the Cell Count would be less than or equal to the SCR cps (6). It is, so the cell is conforming to the SCR and we increment the counter.
- Cell 14 is an Idle Cell.
- Cell 15 is a CLP(0) data cell.
    - **Burst Count = 2**: Since the cell was received at or before the TAT from the previous data cell it's considered part of the same burst. Since the result of incrementing the burst counter would be less than or equal to the MBS (3) it is Burst conforming so the counter is incremented.
    - **Cell Count = 5**: Since the cell was Burst conforming we now see if the result of incrementing the Cell Count would be less than or equal to the SCR cps (6). It is, so the cell is conforming to the SCR and we increment the counter.
- Cell 16 is an Idle Cell.
- Cell 17 is a CLP(0) data cell.
    - **Burst Count = 3**: Since the cell was received at or before the TAT from the previous data cell it's considered part of the same burst. Since the result of incrementing the burst counter would be less than or equal to the MBS (3) it is Burst conforming so the counter is incremented.
    - **Cell Count = 6**: Since the cell was Burst conforming we now see if the result of incrementing the Cell Count would be less than or equal to the SCR cps (6). It is, so the cell is conforming to the SCR and we increment the counter.
- Cell 18 is an Idle Cell.
- Cell 19 is an Idle Cell but it's where the next data call at PCR rate would have been (TAT7). This means the burst containing Cells 13, 15, and 17, is over.
    - **Burst Count = 0**: We reset the Burst Counter to prepare for the next burst.
    - **Cell Count = 6**: We **don't** reset the Cell Count since that keeps track of the total number of cells we can send in one second.

- Cells 20 is an Idle Cell. Note that if cell 20 had been a data cell it would have been the start of a new burst but would have exceeded the SCR cps making it nonconforming. Since Cell 20 represented the last slot for the one second period:
  - **Burst Count = 0:** Even though it was already set to 0 due to Cell 19, we'll do it again to be sure.
  - **Cell Count = 0**: We reset the Cell Count.

Simple, right? Remember the main points are:

- A burst cannot exceed the MBS value.
- Bursts must be separated by at least one PCR TAT which does not contain a PCR/CDVT conforming data cell.
- The total number of conforming data cells can't exceed the SCR value.

Let's try another one.

SCR(0+1)/MBS(0+1) Policing: Example 2

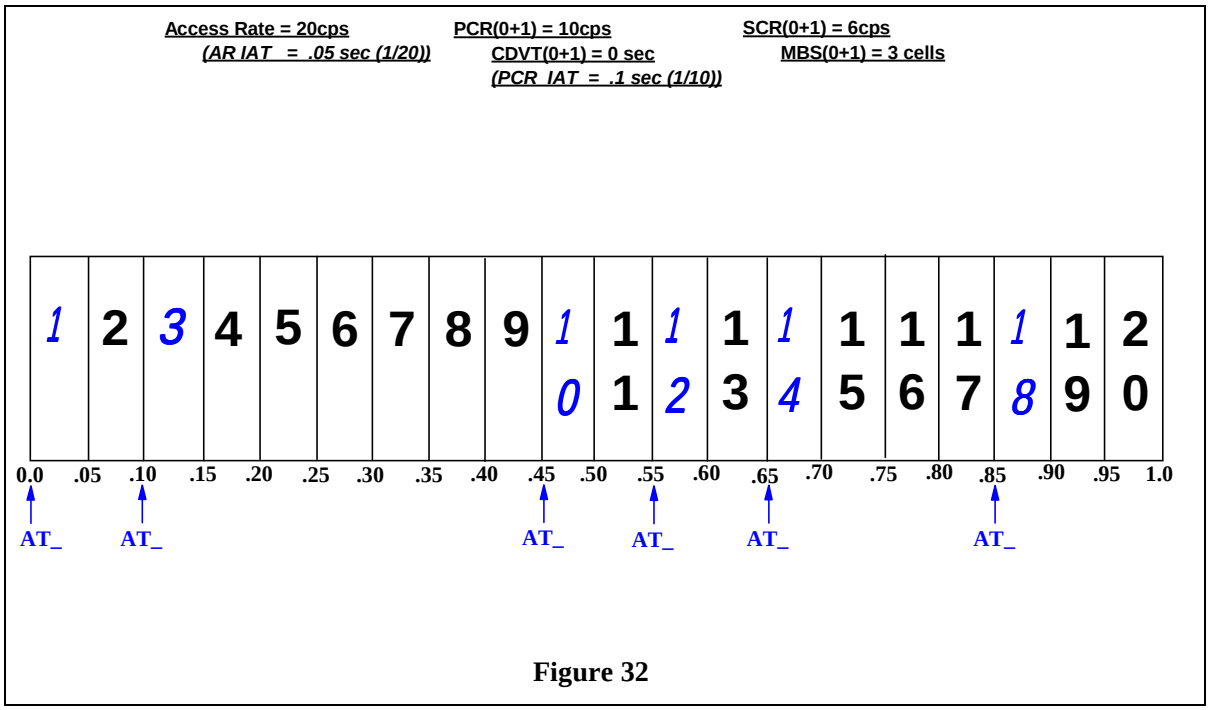

Figure 32

Note that all the traffic parameters are the same as the previous example.

The result of the PCR/CDVT policing is as follows:

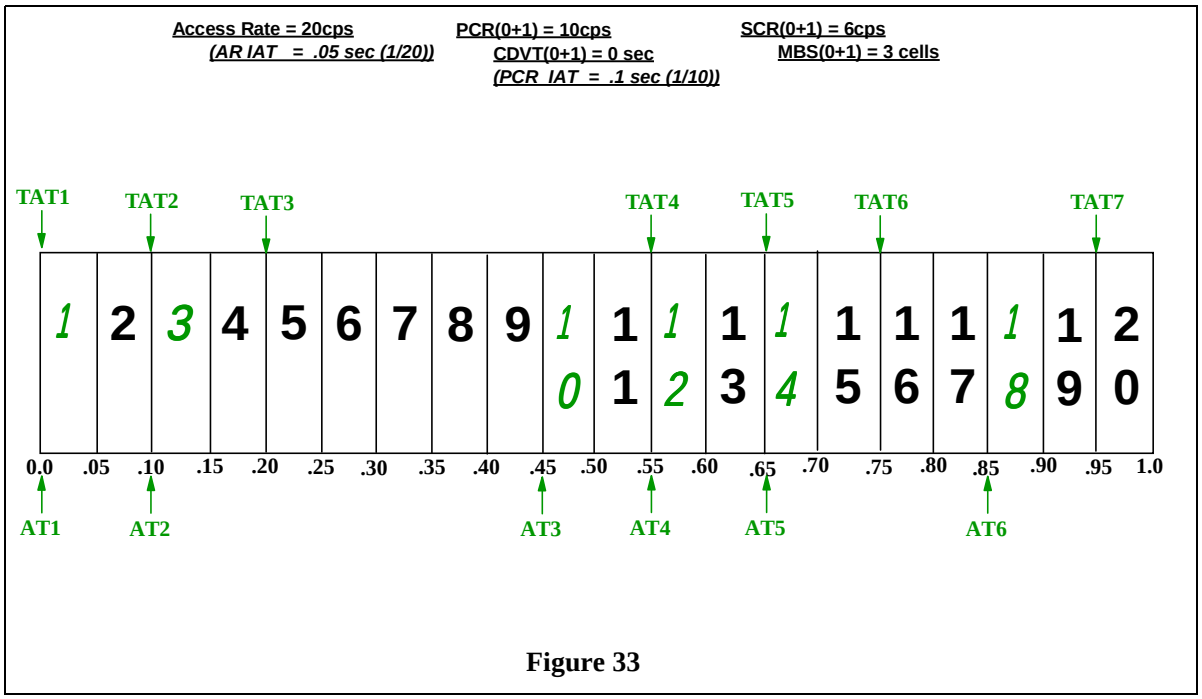

Figure 33

Again, all the data cells were conforming.

Now do the SCR/MBS testing and you should come up with the following:

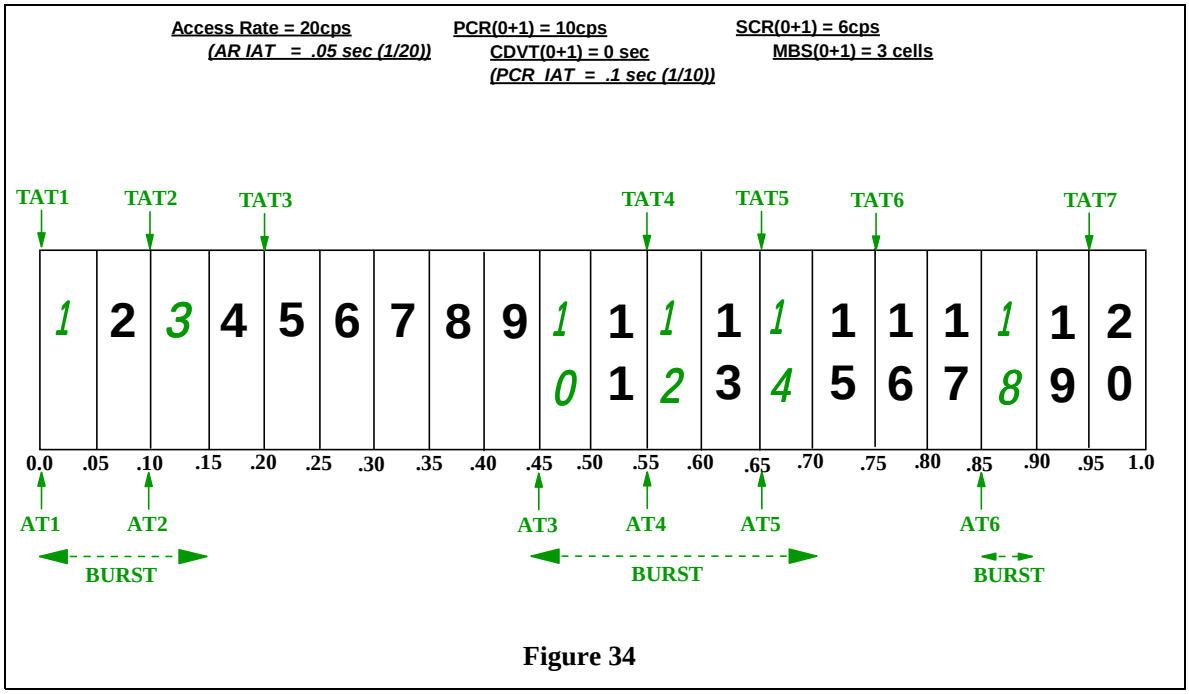

Figure 34

- Cell 1 is a CLP(0) data cell and was received at time .00sec. It is the first data cell as well as the first cell of a burst so:

    - **Burst Count = 1**: This is less than or equal to the MBS (3) so it is Burst conforming.
    - **Cell Count = 1**: This is less than or equal to the SCR cps (6) so it is SCR conforming.
- Cell 2 is an Idle Cell.
- Cell 3 is a CLP(0) data cell.
    - **Burst Count = 2**: Since the cell was received at or before the TAT from the previous data cell it's considered part of the same burst. Since the result of incrementing the burst counter would be less than or equal to the MBS (3) it is Burst conforming so the counter is incremented.
    - **Cell Count = 2**: Since the cell was Burst conforming we now see if the result of incrementing the Cell Count would be less than or equal to the SCR cps (6). It is, so the cell is conforming to the SCR and we increment the counter.
- Cell 4 is an Idle Cell.
- Cell 5 is an Idle Cell but it's where the next data cell at PCR rate would have been (TAT3). This means the burst containing Cells 1 and 3 is over. Remember, MBS stands for Maximum Burst Size. Smaller bursts are ok.
    - **Burst Count = 0**: We reset the Burst Counter to prepare for the next burst.
    - **Cell Count = 2**: We don't reset the Cell Count since that keeps track of the total number of cells we can send in one second.
- Cells 6 through 9 are Idle Cells.
- Cell 10 is a CLP(0) data cell.
    - **Burst Count = 1**: Since there was a PCR TAT *gap* after the last conforming data cell (TAT3) this is the first cell of a new burst. The count is less than or equal to the MBS (3) so it is Burst conforming.
    - **Cell Count = 3**: Since the cell was Burst conforming we now see if the result of incrementing the Cell Count would be less than or equal to the SCR cps (6). It is, so the cell is conforming to the SCR and we increment the counter.
- Cell 11 is an Idle Cell.
- Cell 12 is a CLP(0) data cell.
    - **Burst Count = 2**: Since the cell was received at or before the TAT from the previous data cell it's considered part of the same burst. Since the result of incrementing the burst counter would be less than or equal to the MBS (3) it is Burst conforming so the counter is incremented.
    - **Cell Count = 4**: Since the cell was Burst conforming we now see if the result of incrementing the Cell Count would be less than or equal to the SCR cps (6). It is, so the cell is conforming to the SCR and we increment the counter.
- Cell 13 is an Idle Cell.
- Cell 14 is a CLP(0) data cell.
    - **Burst Count = 3**: Since the cell was received at or before the TAT from the previous data cell it's considered part of the same burst. Since the result of incrementing the burst counter would be less than or equal to the MBS (3) it is Burst conforming so the counter is incremented. Note that we are now at the maximum burst size (3).
    - **Cell Count = 5**: Since the cell was Burst conforming we now see if the result of incrementing the Cell Count would be less than or equal to the SCR cps (6). It is, so the cell is conforming to the SCR and we increment the counter.
- Cell 15 is an Idle Cell.

- Cell 16 is an Idle Cell but it's where the next data cell at PCR rate would have been (TAT6). This means the burst containing Cells 10, 12, and 14 is over.
  - **Burst Count = 0:** We reset the Burst Counter to prepare for the next burst.
  - **Cell Count = 5**: We **don't** reset the Cell Count since that keeps track of the total number of cells we can send in one second.
- Cell 17 is an Idle Cell.
- Cell 18 is a CLP(0) data cell.
  - **Burst Count = 1**: Since there was a PCR TAT *gap* after the last conforming data cell (TAT6) this is the first cell of a new burst. The count is less than or equal to the MBS (3) so it is Burst conforming.
  - **Cell Count = 6**: Since the cell was Burst conforming we now see if the result of incrementing the Cell Count would be less than or equal to the SCR cps (6). It is, so the cell is conforming to the SCR and we increment the counter.
- Cell 19 is an Idle Cell.
- Cells 20 is an Idle Cell. If it had been a data cell it would have been considered part of Cell 18's burst and would have been Burst conforming but would have caused SCR to be exceeded so it would have been deemed nonconforming and discarded. Since it's an Idle Cell the previous burst is over after only one cell. Since Cell 20 represented the last slot for the one second period:
  - **Burst Count = 0:** Even though it was already set to 0 due to Cell 19, we'll do it again to be sure.
  - **Cell Count = 0**: We reset the Cell Count.

Now let's look at a few scenarios containing naughty cells.

SCR(0+1)/MBS(0+1) Policing: Example 3

Note that all the traffic parameters are the same as in the previous example.

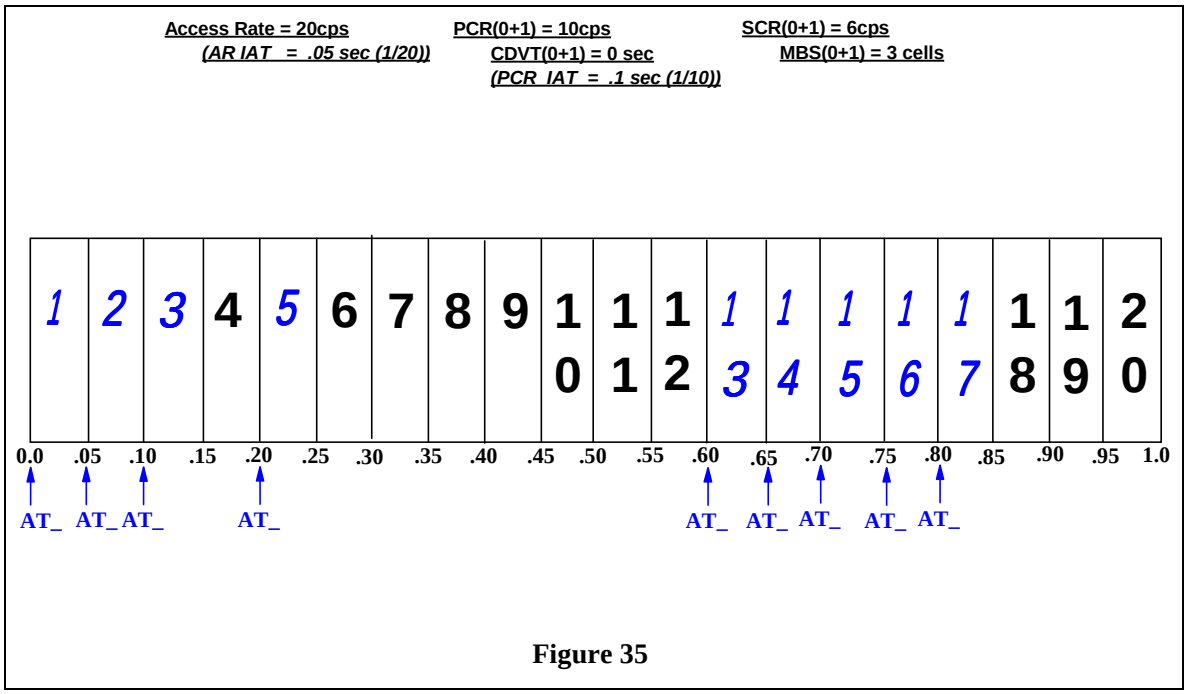

Figure 35

The result of the PCR/CDVT policing is shown on the next page.

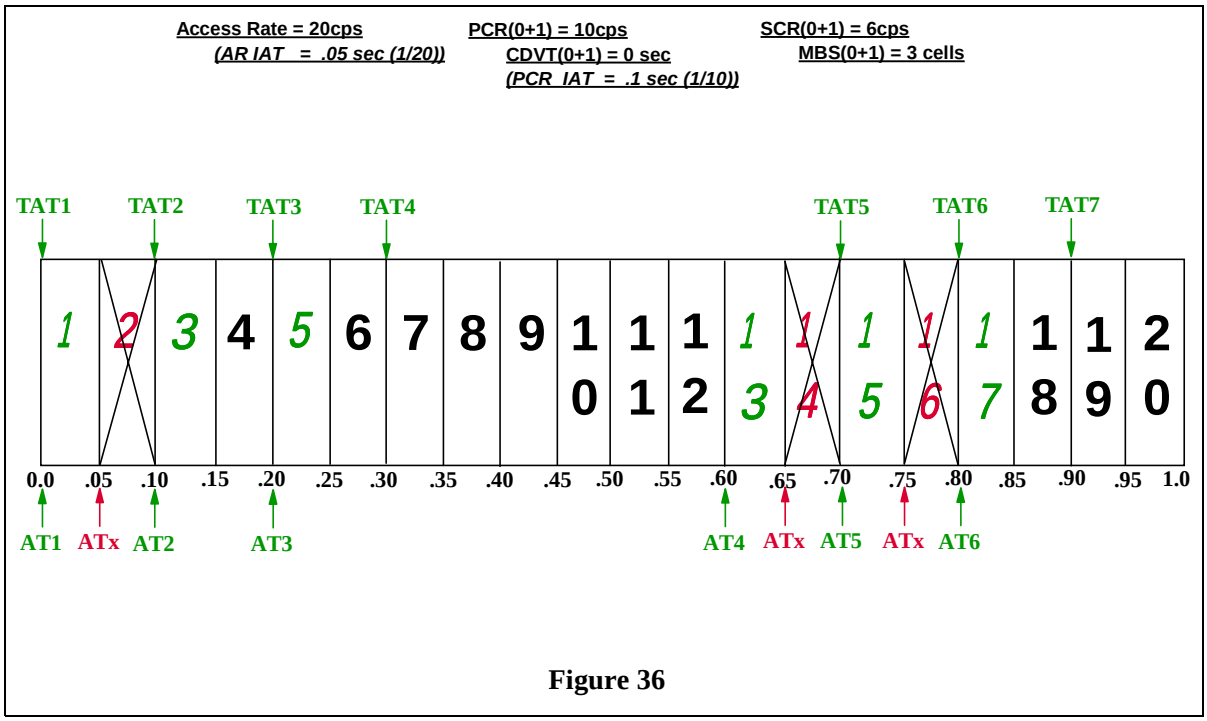

Figure 36

Cells 2, 14, and 16, were nonconforming to the PCR/CDVT testing.

Now use this as input for the SCR/MBS testing and you should come up with the following:

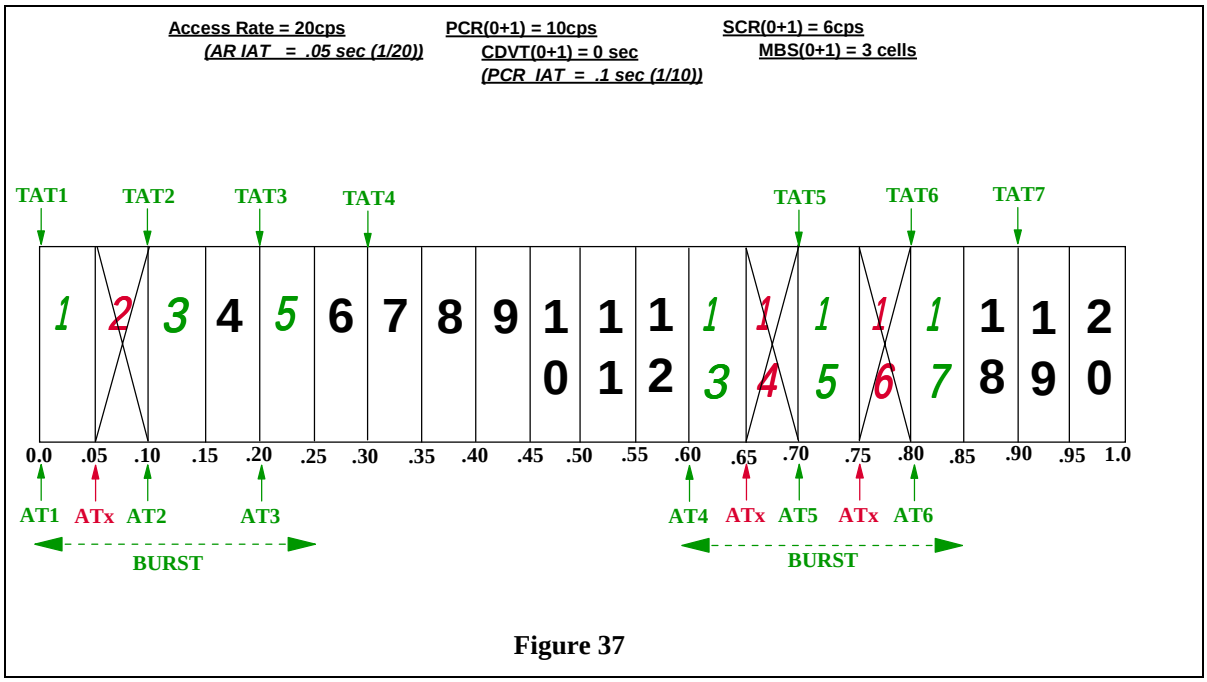

Figure 37

- Cell 1 is a CLP(0) data cell and was received at time .00sec. It is the first data cell as well as the first cell of a burst so:
  - **Burst Count = 1**: This is less than or equal to the MBS (3) so it is Burst conforming.
  - **Cell Count = 1**: This is less than or equal to the SCR cps (6) so it is SCR conforming.
- Cell 2 is a CLP(0) data cell deemed nonconforming by the PCR/CDVT test and discarded so its time slot in the stream is treated the same as an Idle Cell.
- Cell 3 is a CLP(0) data cell.
  - **Burst Count = 2**: Since the cell was received at or before the TAT from the previous data cell it's considered part of the same burst. Since the result of incrementing the burst counter would be less than or equal to the MBS (3) it is Burst conforming so the counter is incremented.
  - **Cell Count = 2**: Since the cell was Burst conforming we now see if the result of incrementing the Cell Count would be less than or equal to the SCR cps (6). It is, so the cell is conforming to the SCR and we increment the counter.
- Cell 4 is an Idle Cell.
- Cell 5 is a CLP(0) data cell.
  - **Burst Count = 3**: Since the cell was received at or before the TAT from the previous data cell it's considered part of the same burst. Since the result of incrementing the burst counter would be less than or equal to the MBS (3) it is Burst conforming so the counter is incremented.
  - **Cell Count = 3**: Since the cell was Burst conforming we now see if the result of incrementing the Cell Count would be less than or equal to the SCR cps (6). It is, so the cell is conforming to the SCR and we increment the counter.
- Cell 6 is an Idle Cell.
- Cell 7 is an Idle Cell but it's where the next data cell at PCR rate would have been (TAT4). This means the burst containing Cells 1 , 3 and 5 is over.
  - **Burst Count = 0**: The Burst Counter is reset to prepare for the next burst.
  - **Cell Count = 3**: The cell count is not reset since it keeps track of the total number of cells that can be sent in one second.
- Cells 8 through 12 are Idle Cells.
- Cell 13 is a CLP(0) data cell.
  - **Burst Count = 1**: Since there was a PCR TAT *gap* after the last conforming data cell (TAT4) this is the first cell of a new burst. The count is less than or equal to the MBS (3) so it is Burst conforming.
  - **Cell Count = 4**: Since the cell was Burst conforming we now see if the result of incrementing the Cell Count would be less than or equal to the SCR cps (6). It is, so the cell is conforming to the SCR and we increment the counter.
- Cell 14 is a CLP(0) data cell deemed nonconforming by the PCR/CDVT test and discarded so its time slot in the stream is treated the same as an Idle Cell.
- Cell 15 is a CLP(0) data cell.
  - **Burst Count = 2**: Since the cell was received at or before the TAT from the previous data cell it's considered part of the same burst. Since the result of incrementing the burst counter would be less than or equal to the MBS (3) it is Burst conforming so the counter is incremented.

    - **Cell Count = 5**: Since the cell was Burst conforming we now see if the result of incrementing the Cell Count would be less than or equal to the SCR cps (6). It is, so the cell is conforming to the SCR and we increment the counter.
- Cell 16 is a CLP(0) data cell deemed nonconforming by the PCR/CDVT test and discarded so its time slot in the stream is treated the same as an Idle Cell.
- Cell 17 is a CLP(0) data cell.
    - **Burst Count = 3**: Since the cell was received at or before the TAT from the previous data cell it's considered part of the same burst. Since the result of incrementing the burst counter would be less than or equal to the MBS (3) it is Burst conforming so the counter is incremented.
    - **Cell Count = 6**: Since the cell was Burst conforming we now see if the result of incrementing the Cell Count would be less than or equal to the SCR cps (6). It is, so the cell is conforming to the SCR and we increment the counter.
- Cell 18 is an Idle Cell.
- Cell 19 is an Idle Cell but it's where the next data cell at PCR rate would have been (TAT7). This means the burst containing Cells 13 , 15 and 17 is over.
    - **Burst Count = 0**: The Burst Counter is reset to prepare for the next burst.
    - **Cell Count = 6**: The cell count is not reset since it keeps track of the total number of cells that can be sent in one second.
- Cell 20 is an Idle Cell. Since it represents the last slot for the one second period:
    - **Burst Count = 0**: The Burst Counter is reset to prepare for the next burst.
    - **Cell Count = 0**: The Cell Counter is also reset.

So in this example the only nonconformance occurred in the PCR/CDVT test.

Let's try another one.

SCR(0+1)/MBS(0+1) Policing: Example 4

All the traffic parameters are the same as the previous example.

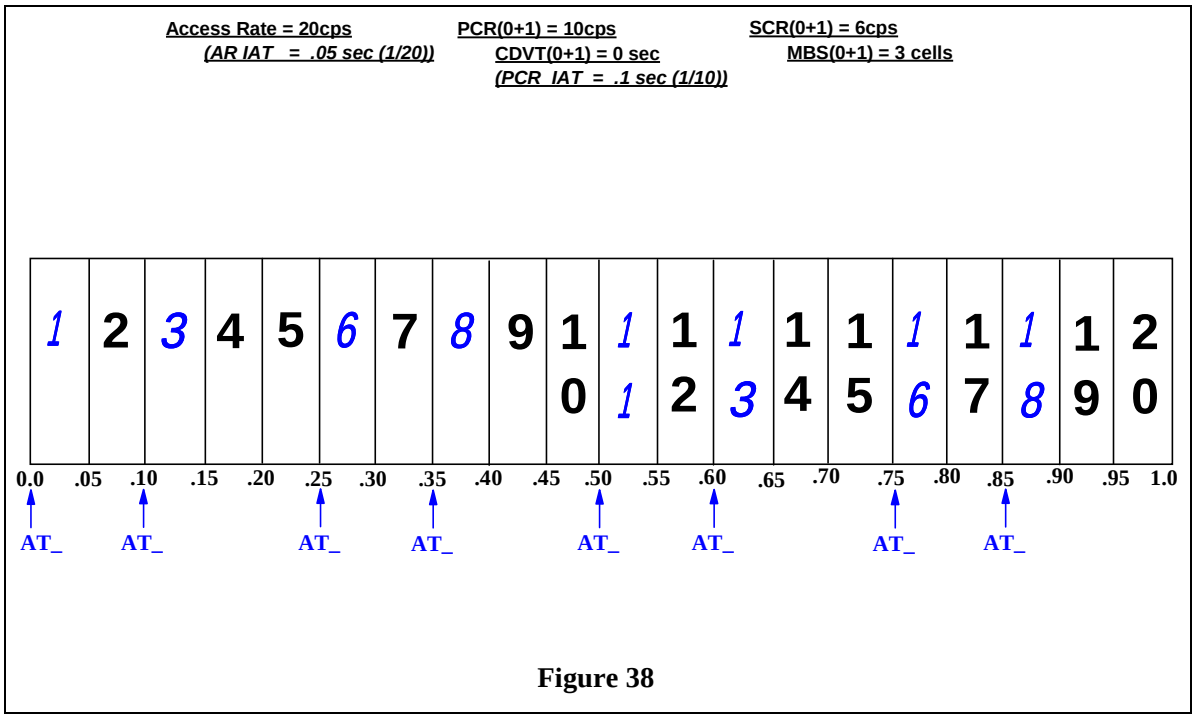

Figure 38

The result of the PCR/CDVT policing is on the next page.

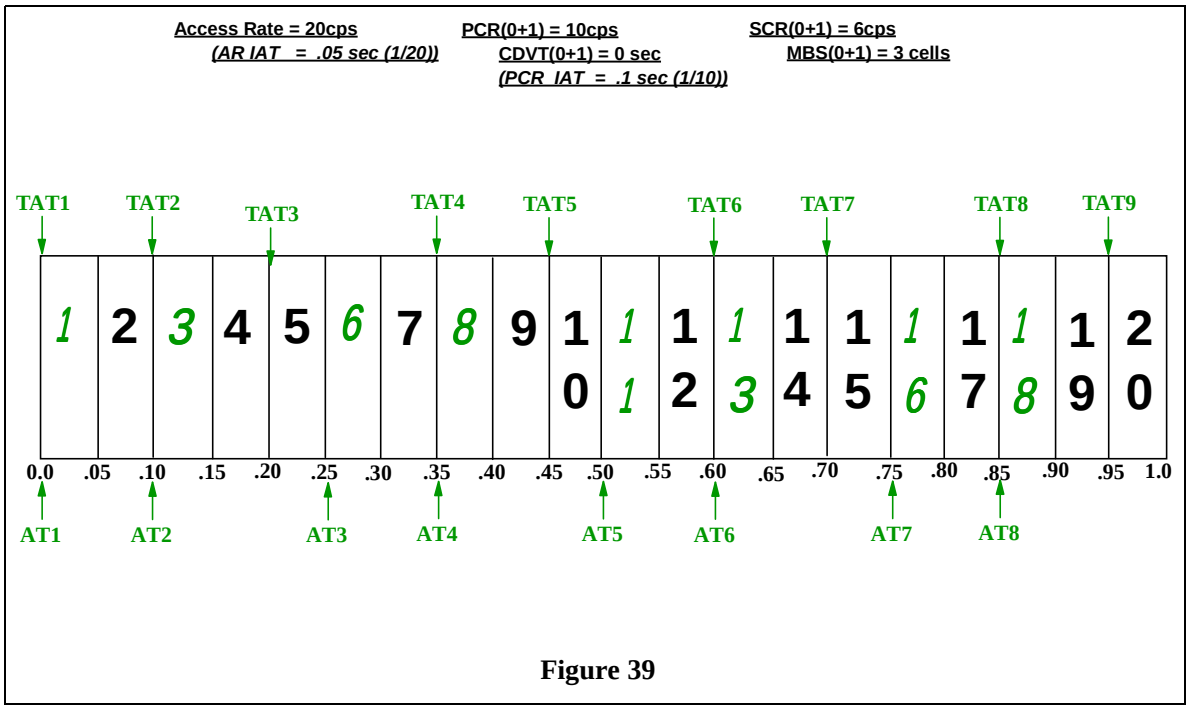

Figure 39

All the cells were conforming at PCR/CDVT.

Now test for SCR/MBS conformance.

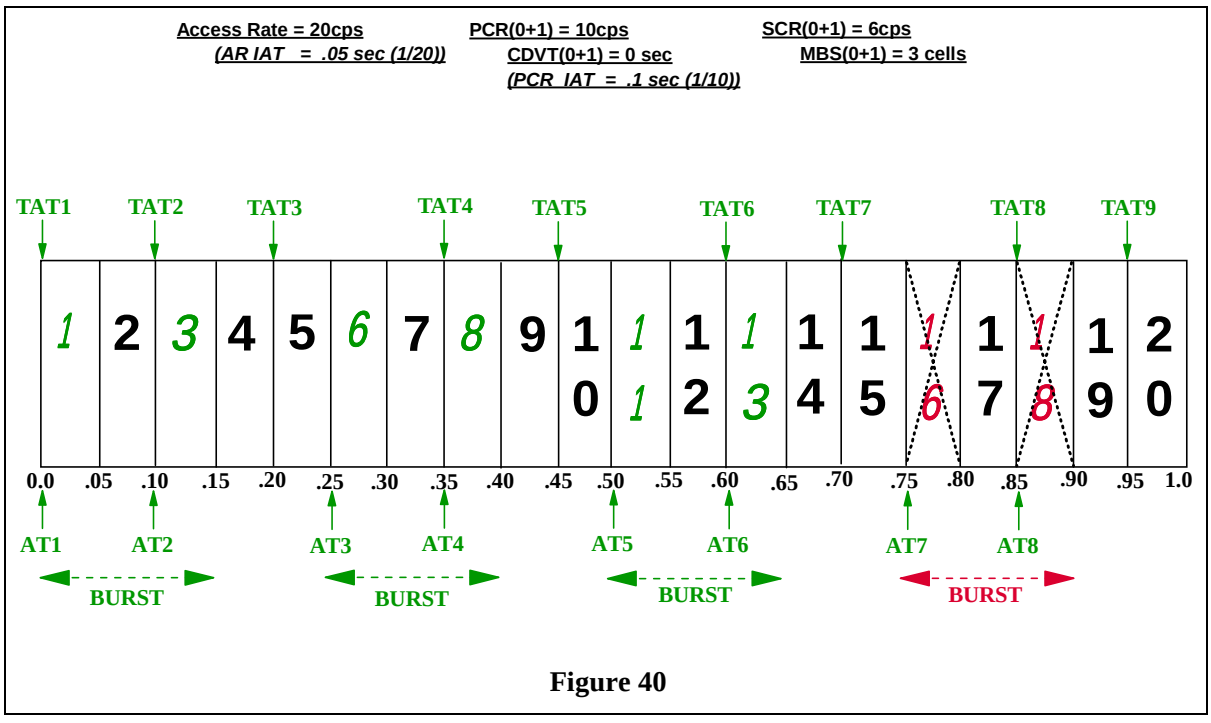

Figure 40

- Cell 1 is a CLP(0) data cell and was received at time .00sec. It is the first data cell as well as the first cell of a burst so:
  - **Burst Count = 1**: This is less than or equal to the MBS (3) so it is Burst conforming.
  - **Cell Count = 1**: This is less than or equal to the SCR cps (6) so it is SCR conforming.
- Cell 2 is an Idle Cell.
- Cell 3 is a CLP(0) data cell.
  - **Burst Count = 2**: Since the cell was received at or before the TAT from the previous data cell it's considered part of the same burst. Since the result of incrementing the burst counter would be less than or equal to the MBS (3) it is Burst conforming so the counter is incremented.
  - **Cell Count = 2**: Since the cell was Burst conforming we now see if the result of incrementing the Cell Count would be less than or equal to the SCR cps (6). It is, so the cell is conforming to the SCR and we increment the counter.
- Cell 4 is an Idle Cell.
- Cell 5 is an Idle Cell but it's where the next data cell at PCR rate would have been (TAT3). This means the burst containing Cells 1 and 3 is over.
  - **Burst Count = 0**: The Burst Counter is reset to prepare for the next burst.
  - **Cell Count = 2**: The cell count is not reset since it keeps track of the total number of cells that can be sent in one second.
- Cell 6 is a CLP(0) data cell.
  - **Burst Count = 1**: Since there was a PCR TAT *gap* after the last conforming data cell (TAT3) this is the first cell of a new burst. The count is less than or equal to the MBS (3) so it is Burst conforming.
  - **Cell Count = 3**: Since the cell was Burst conforming we now see if the result of incrementing the Cell Count would be less than or equal to the SCR cps (6). It is, so the cell is conforming to the SCR and we increment the counter.
- Cell 7 is an Idle Cell.
- Cell 8 is a CLP(0) data cell and was received at time .35sec.:
  - **Burst Count = 2**: Since the cell was received at or before the TAT from the previous data cell it's considered part of the same burst. Since the result of incrementing the burst counter would be less than or equal to the MBS (3) it is Burst conforming so the counter is incremented.
  - **Cell Count = 4**: Since the cell was Burst conforming we now see if the result of incrementing the Cell Count would be less than or equal to the SCR cps (6). It is, so the cell is conforming to the SCR and we increment the counter.
- Cell 9 is an Idle Cell.
- Cell 10 is an Idle Cell but it's where the next data cell at PCR rate would have been (TAT5). This means the burst containing Cells 6 and 8 is over.
  - **Burst Count = 0**: The Burst Counter is reset to prepare for the next burst.
  - **Cell Count = 4**: The cell count is not reset since it keeps track of the total number of cells that can be sent in one second.
- Cell 11 is a CLP(0) data cell.
  - **Burst Count = 1**: Since there was a PCR TAT *gap* after the last conforming data cell (TAT5) this is the first cell of a new burst. The count is less than or equal to the MBS (3) so it is Burst conforming.

    - **Cell Count = 5**: Since the cell was Burst conforming we now see if the result of incrementing the Cell Count would be less than or equal to the SCR cps (6). It is, so the cell is conforming to the SCR and we increment the counter.
- Cell 12 is an Idle Cell.
- Cell 13 is a CLP(0) data cell.
    - **Burst Count = 2**: Since the cell was received at or before the TAT from the previous data cell it's considered part of the same burst. Since the result of incrementing the burst counter would be less than or equal to the MBS (3) it is Burst conforming so the counter is incremented.
    - **Cell Count = 6**: Since the cell was Burst conforming we now see if the result of incrementing the Cell Count would be less than or equal to the SCR cps (6). It is, so the cell is conforming to the SCR and we increment the counter. **Note that we are now at the SCR limit.**
- Cell 14 is an Idle Cell.
- Cell 15 is an Idle Cell but it's where the next data cell at PCR rate would have been (TAT7). This means the burst containing Cells 11 and 13 is over.
    - **Burst Count = 0**: The Burst Counter is reset to prepare for the next burst.
    - **Cell Count = 6**: The cell count is not reset since it keeps track of the total number of cells that can be sent in one second.
- Cell 16 is a CLP(0) data cell.
    - **Burst Count = 1**: Since there was a PCR TAT *gap* after the last conforming data cell (TAT7) this is the first cell of a new burst. The count is less than or equal to the MBS (3) so it is Burst conforming.
    - **Cell Count = 6**: Since the cell was Burst conforming we now see if the result of incrementing the Cell Count would be less than or equal to the SCR cps (6). **The result would be greater than the SCR cps so the Cell Count stays at 6.** Since this also makes the cell nonconforming, it will be discarded.
    - **Burst Count = 0**: Oops! This count needs to be reset since the cell really wasn't conforming. I warned you this would be a rough description of how it works.
- Cell 17 is an Idle Cell.
- Cell 18 is a CLP(0) data cell.
    - **Burst Count = 1**: Since there was a PCR TAT *gap* after the last conforming data cell (TAT7) this is the first cell of a new burst. The count is less than or equal to the MBS (3) so it is Burst conforming.
    - **Cell Count = 6**: Since the cell was Burst conforming we now see if the result of incrementing the Cell Count would be less than or equal to the SCR cps (6). **The result would be greater than the SCR cps so the Cell Count stays at 6.** Since this also makes the cell nonconforming, it will be discarded.
    - **Burst Count = 0**: This count needs to be reset since the cell really wasn't conforming.
- Cell 19 is an Idle Cell.
- Cell 20 is an Idle Cell. Since it represents the last slot for the one second period:
    - **Burst Count = 0**: The Burst Counter is reset to prepare for the next burst.
    - **Cell Count = 0**: The Cell Counter is reset.

So all the cells passed PCR/CDVT and all bursts were within MBS. Unfortunately the SCR was reached with the third burst so all data cells after that were nonconforming.

Up to now there have been failures at PCR and MBS. Can you guess the next scenario?

SCR(0+1)/MBS(0+1) Policing: Example 5

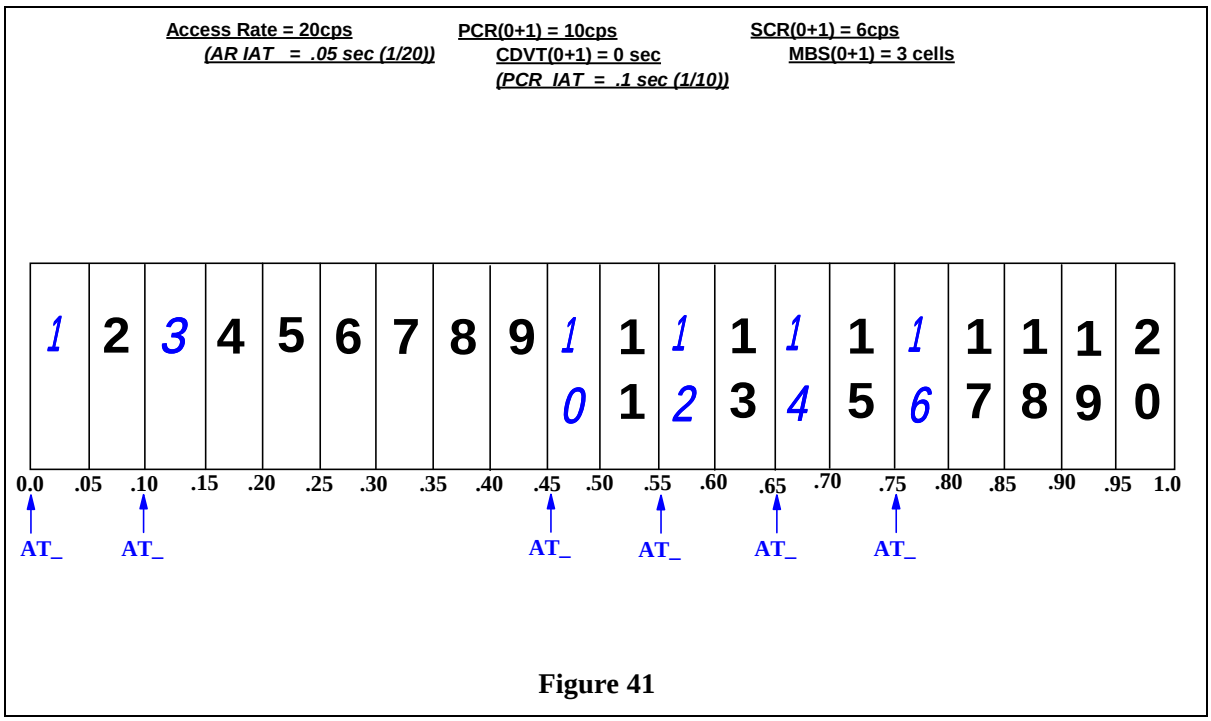

Figure 41

All the traffic parameters are the same as the previous example.

The result of the PCR/CDVT policing is on the next page.

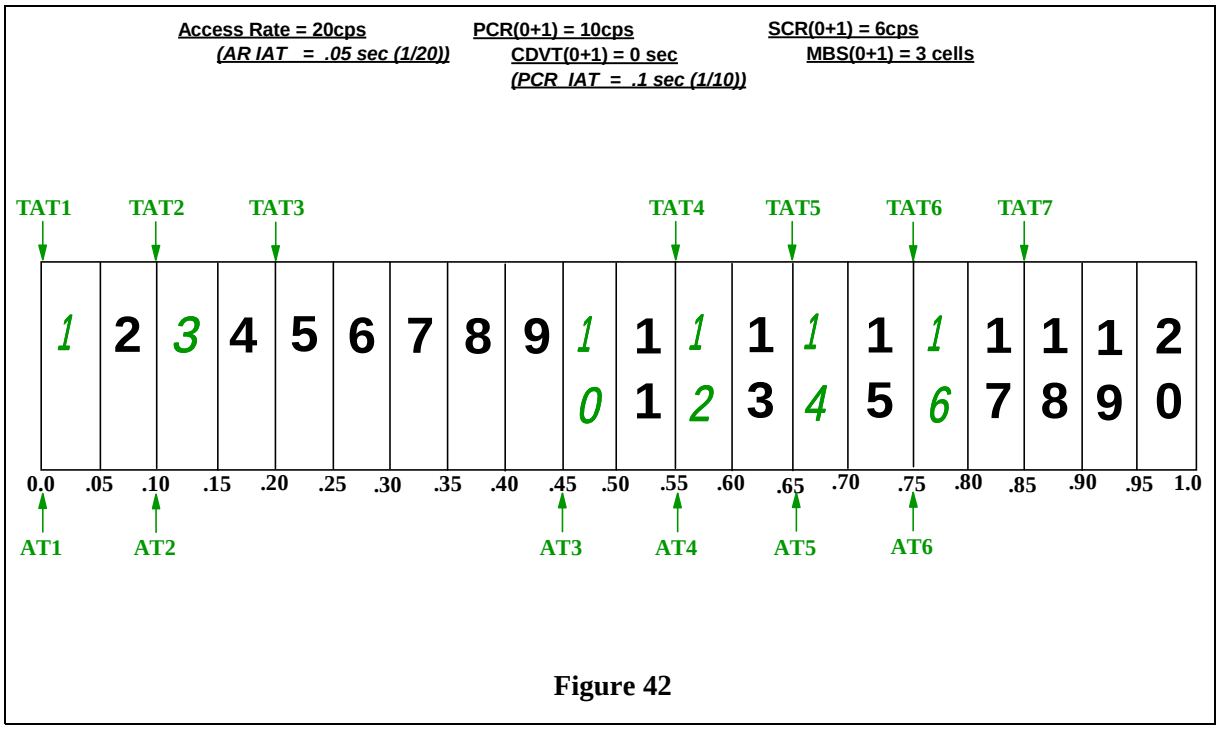

Figure 42

All the cells were conforming at PCR/CDVT.

Now test for SCR/MBS conformance.

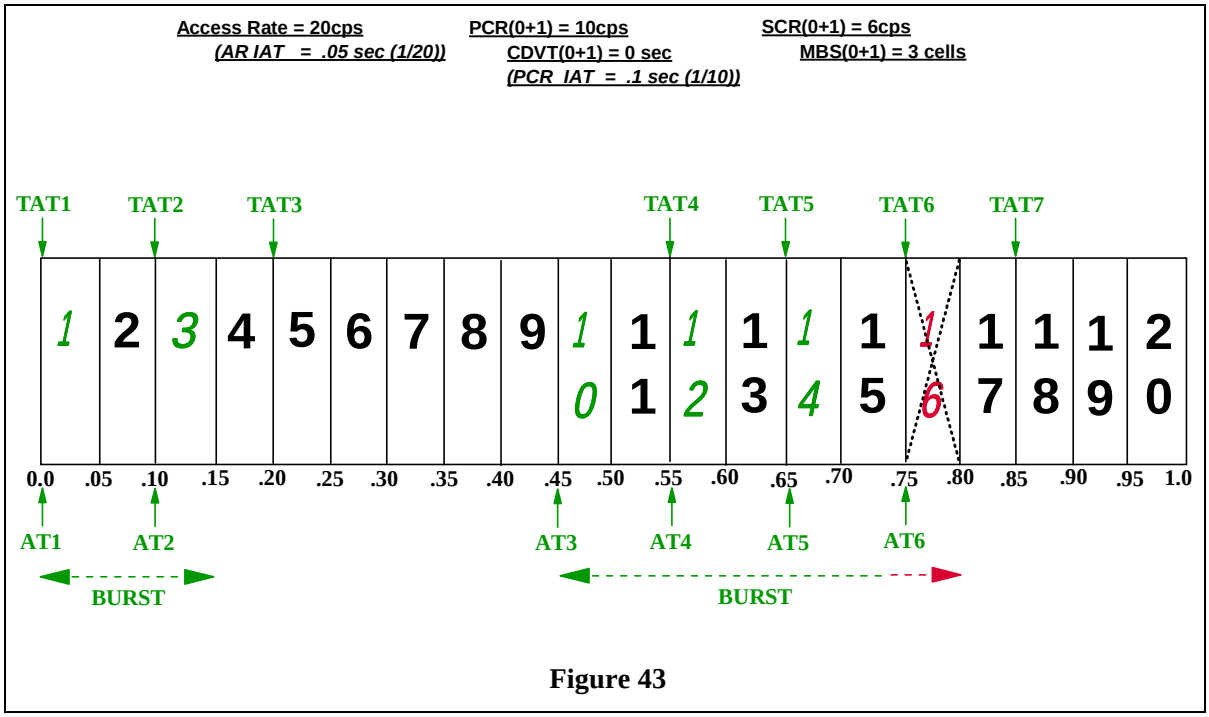

Figure 43

- Cell 1 is a CLP(0) data cell and was received at time .00sec. It is the first data cell as well as the first cell of a burst so:
  - **Burst Count = 1**: This is less than or equal to the MBS (3) so it is Burst conforming.
  - **Cell Count = 1**: This is less than or equal to the SCR cps (6) so it is SCR conforming.
- Cell 2 is an Idle Cell.
- Cell 3 is a CLP(0) data cell.
  - **Burst Count = 2**: Since the cell was received at or before the TAT from the previous data cell it's considered part of the same burst. Since the result of incrementing the burst counter would be less than or equal to the MBS (3) it is Burst conforming so the counter is incremented.
  - **Cell Count = 2**: Since the cell was Burst conforming we now see if the result of incrementing the Cell Count would be less than or equal to the SCR cps (6). It is, so the cell is conforming to the SCR and we increment the counter.
- Cell 4 is an Idle Cell.
- Cell 5 is an Idle Cell but it's where the next conforming data cell at PCR rate would have been (TAT3). This means the burst containing Cells 1 and 3 is over.
  - **Burst Count = 0**: The Burst Counter is reset to prepare for the next burst.
  - **Cell Count = 2**: The cell count is not reset since it keeps track of the total number of cells that can be sent in one second.
- Cells 6 through 9 are Idle Cells.
- Cell 10 is a CLP(0) data cell.
  - **Burst Count = 1**: Since there was a PCR TAT *gap* after the last conforming data cell (TAT3) this is the first cell of a new burst. The count is less than or equal to the MBS (3) so it is Burst conforming.
  - **Cell Count = 3**: Since the cell was Burst conforming we now see if the result of incrementing the Cell Count would be less than or equal to the SCR cps (6). It is, so the cell is conforming to the SCR and we increment the counter.
- Cell 11 is an Idle Cell.
- Cell 12 is a CLP(0) data cell.
  - **Burst Count = 2**: Since the cell was received at or before the TAT from the previous data cell it's considered part of the same burst. Since the result of incrementing the burst counter would be less than or equal to the MBS (3) it is Burst conforming so the counter is incremented.
  - **Cell Count = 4**: Since the cell was Burst conforming we now see if the result of incrementing the Cell Count would be less than or equal to the SCR cps (6). It is, so the cell is conforming to the SCR and we increment the counter.
- Cell 13 is an Idle Cell.
- Cell 14 is a CLP(0) data cell.
  - **Burst Count = 3**: Since the cell was received at or before the TAT from the previous data cell it's considered part of the same burst. Since the result of incrementing the burst counter would be less than or equal to the MBS (3) it is Burst conforming so the counter is incremented.
  - **Cell Count = 5**: Since the cell was Burst conforming we now see if the result of incrementing the Cell Count would be less than or equal to the SCR cps (6). It is, so the cell is conforming to the SCR and we increment the counter.

- Cell 15 is an Idle Cell.
- Cell 16 is a CLP(0) data cell. This is a good one!
  - **Burst Count = 0**:
    - Since the cell was received at or before the TAT from the previous data cell it's considered part of the same burst.
    - **Since the result of incrementing the burst counter would make it greater than the MBS (3) it is not Burst conforming and the cell is discarded.**
    - Since a **conforming** cell at PCR rate was **not** received at the next TAT (TAT6-Cell 16's time) the burst containing Cells 10, 12, and 14 is over so the counter is reset to 0 to prepare for the next burst. **So from the perspective of providing a "gap" between bursts, an SCR or MBS nonconforming cell is just like an Idle Cell.**
  - **Cell Count = 5**: Since the cell was **not** Burst conforming the Cell Count is **not** incremented.
- Cells 17, 18 and 19 are Idle Cells.
- Cell 20 is an Idle Cell. Since it represented the last slot for the one second period:
  - **Burst Count = 0:** The Burst Counter is reset to prepare for the next burst.
  - **Cell Count = 0:** The Cell Counter is reset.

The total number of cells we tried to send at SCR/MBS was 6, the second burst exceeded MBS so Cell 16 was discarded. Only 5 cells were allowed to pass.

Here's one more in the SCR(0+1)/MBS(0+1) category. You should find discards due to PCR/CDVT, SCR, and MBS.

SCR(0+1)/MBS(0+1) Policing: Example 6

All the traffic parameters are the same as the previous example.

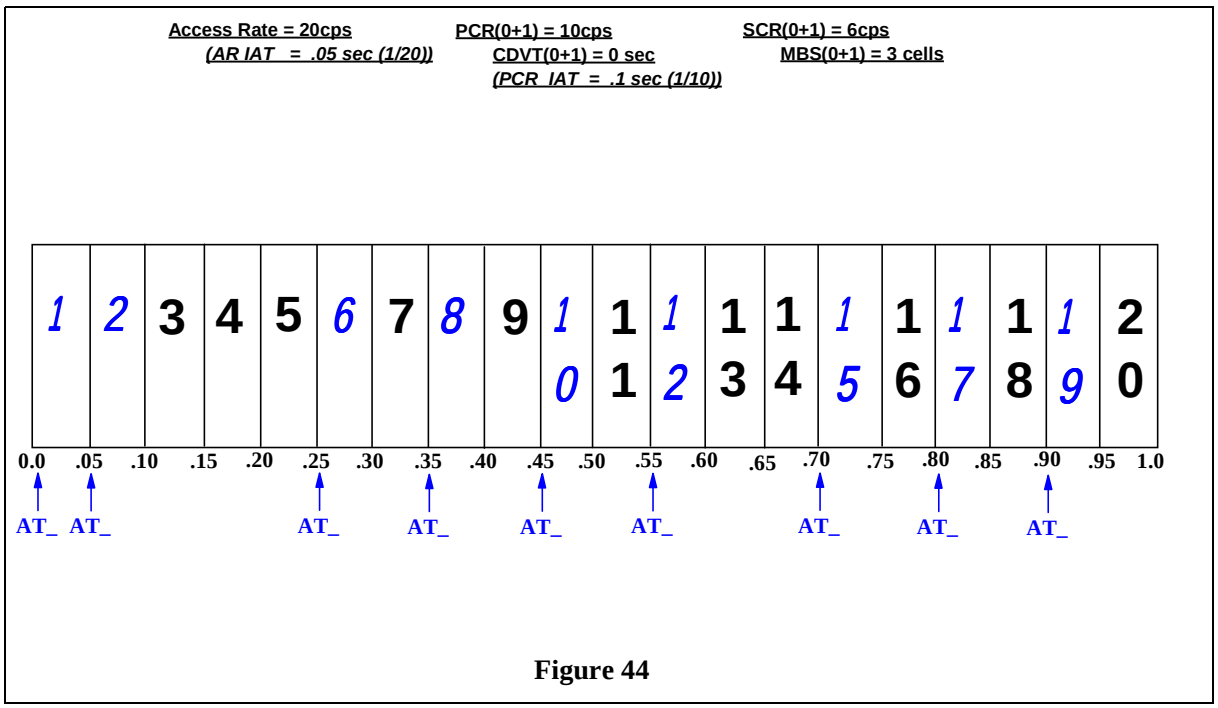

Figure 44

The result of the PCR/CDVT policing is on the next page.

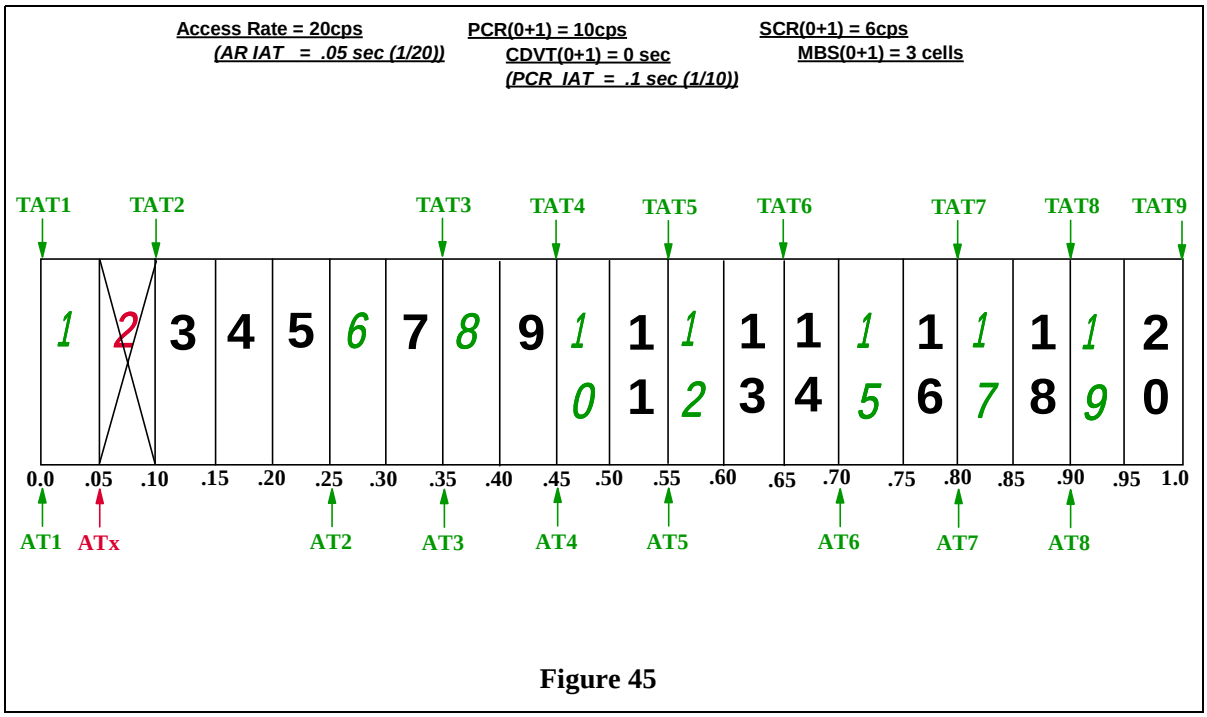

Figure 45

Cell 2 was nonconforming at PCR/CDVT because it came in earlier than TAT2 (and CDVT=0).

Now test for SCR/MBS conformance.

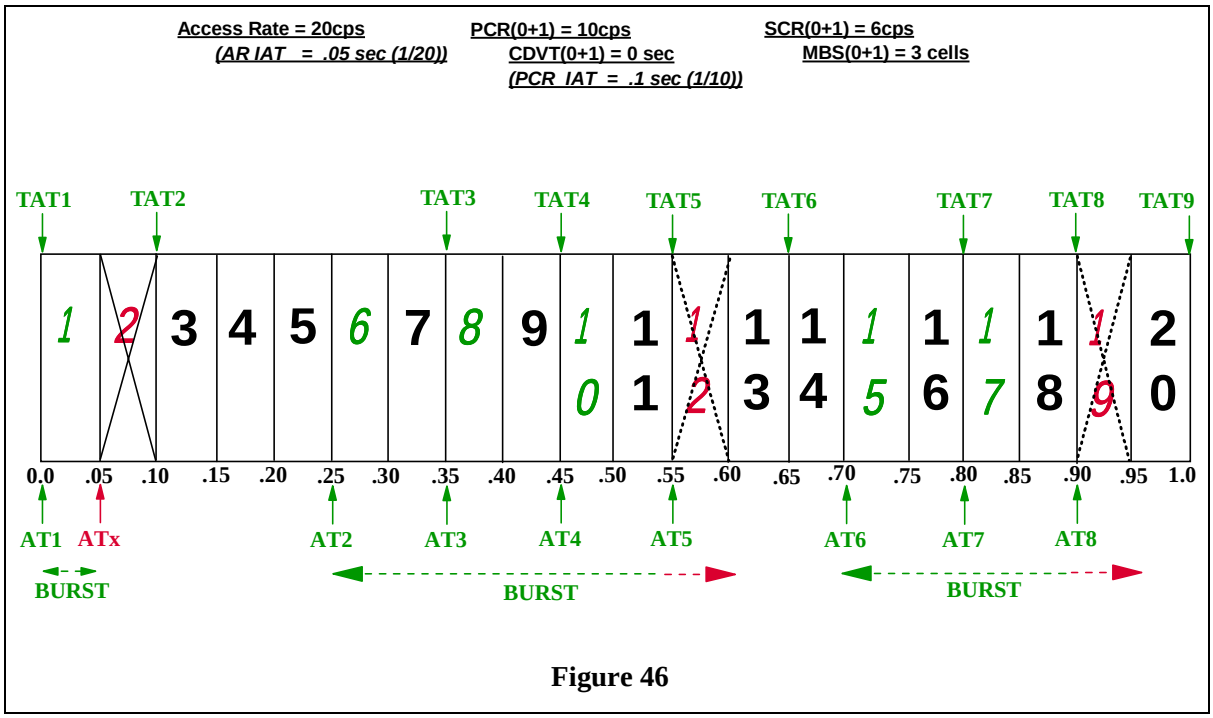

Figure 46

Let’s skip the cell by cell here since this example is so much like the previous ones.

- Cell 12 was discarded because it exceeded the MBS.
- Cell 19 was discarded because it exceeded the SCR cps.

Now let’s try an example in which conformance in the First stage due to the CDVT value has an impact in the Second stage.

SCR(0+1)/MBS(0+1) Policing: Example 7

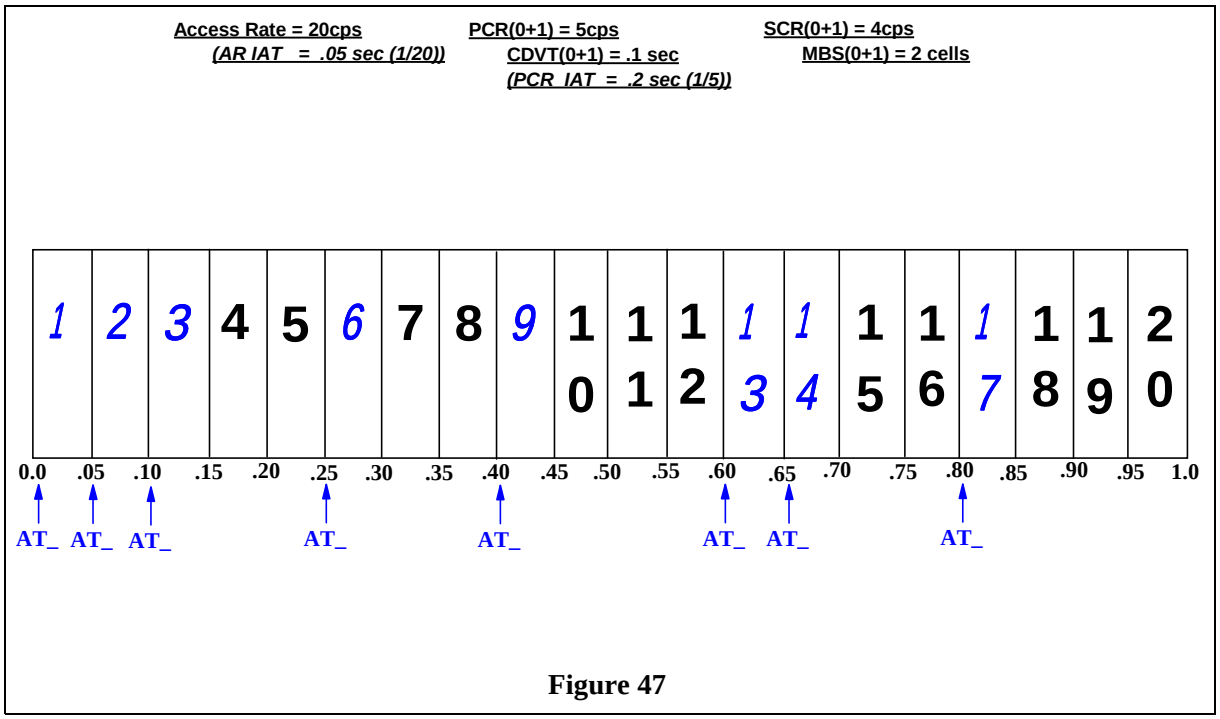

Figure 47

Note that the traffic descriptors are different from the previous examples (especially that CDVT is not 0).

The PCR/CDVT policing should result in the following.

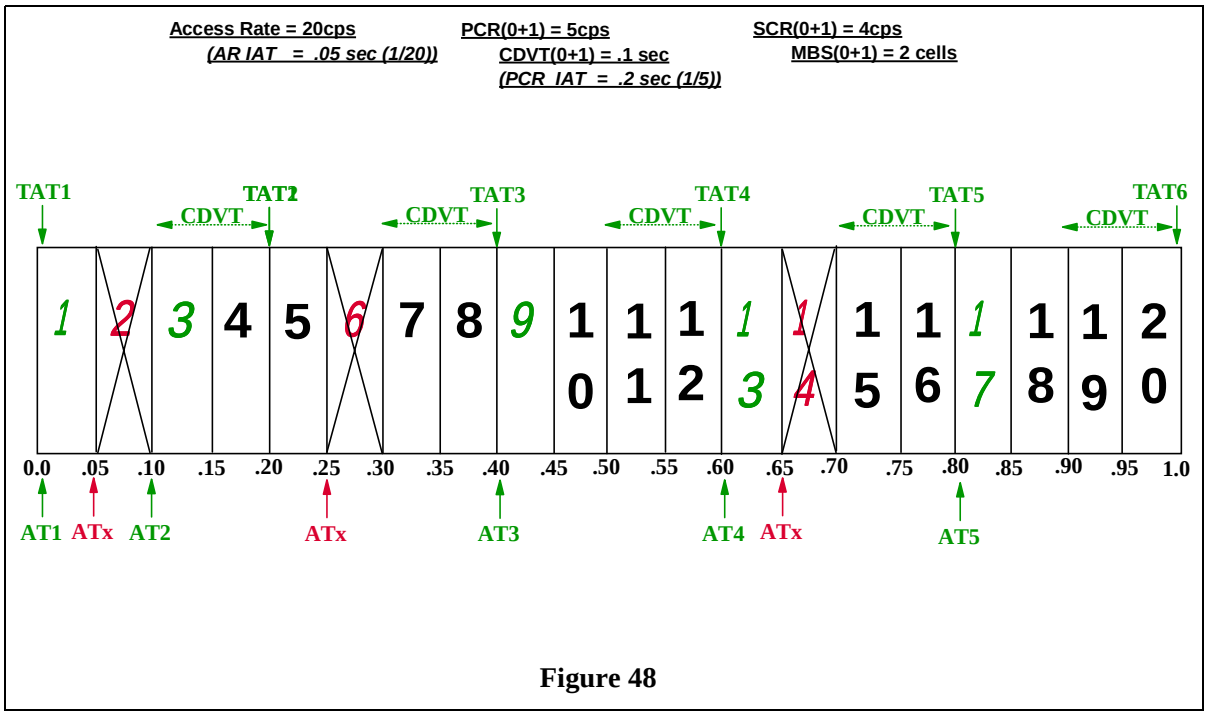

Figure 48

Cells 2, 6, and 14 were nonconforming at PCR/CDVT because they came in earlier than the TATs and CDVTs. Note that Cell 3 was conforming only because it was within the CDVT range.

Now test for SCR/MBS conformance.

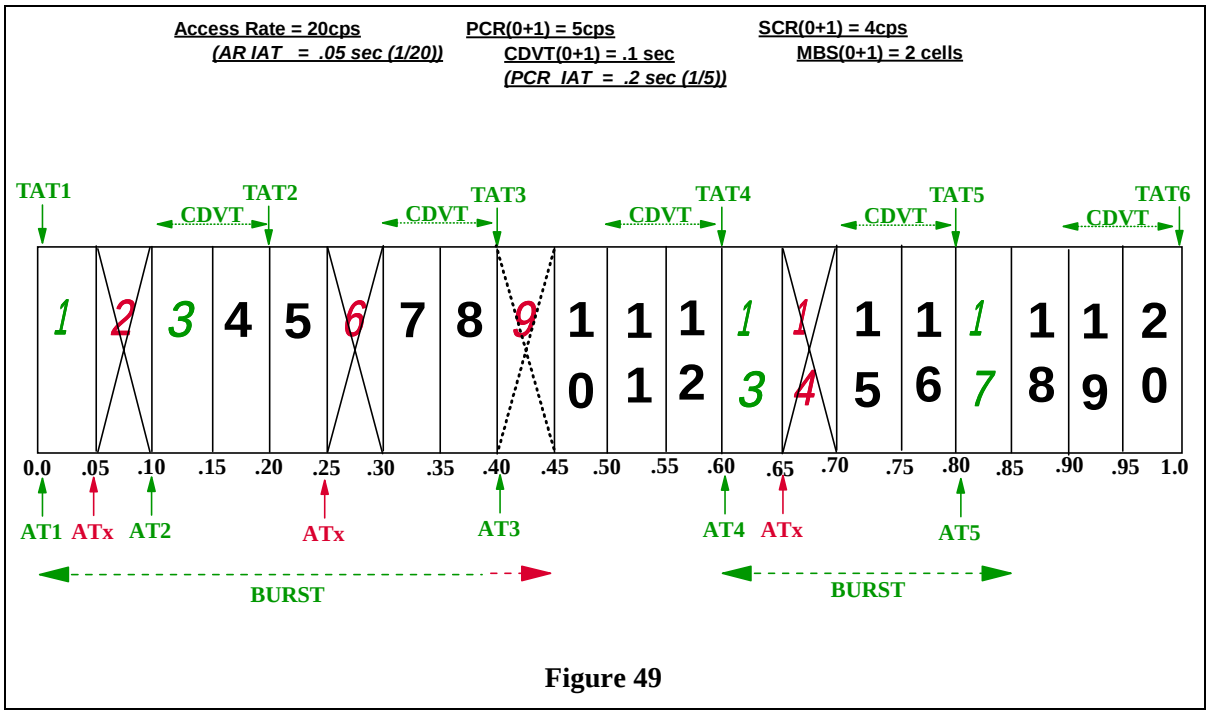

Figure 49

- Cell 1 is a CLP(0) data cell and was received at time .00sec. It is the first data cell as well as the first cell of a burst so:
  - **Burst Count = 1**: This is less than or equal to the MBS (2) so it is Burst conforming.
  - **Cell Count = 1**: This is less than or equal to the SCR cps (4) so it is SCR conforming.
- Cell 2 is a CLP(0) data cell deemed nonconforming by the PCR/CDVT test and discarded so its time slot in the stream is treated the same as an Idle Cell.
- Cell 3 is a CLP(0) data cell.
  - **Burst Count = 2**: Since the cell was received at **or before** the TAT from the previous data cell it's considered part of the same burst. **Note that this cell was conforming to the First Stage only because it was within the CDVT.** Since the result of incrementing the burst counter would be less than or equal to the MBS (2) it is Burst conforming so the counter is incremented.
  - **Cell Count = 2**: Since the cell was Burst conforming we now see if the result of incrementing the Cell Count would be less than or equal to the SCR cps (4). It is, so the cell is conforming to the SCR and we increment the counter.
- Cells 4 and 5 are Idle Cells.
- Cell 6 is a CLP(0) data cell deemed nonconforming by the PCR/CDVT test and discarded so its time slot in the stream is treated the same as an Idle Cell.
- Cells 7 and 8 are Idle Cells.
- Cell 9 is a CLP(0) data cell.
  - **Burst Count = 0**:
    - Since the cell was received at or before the TAT from the previous data cell (TAT3) it's considered part of the same burst.
    - **Since the result of incrementing the burst counter would make it greater than the MBS (2) it is not Burst conforming and the cell is discarded.**
    - Since a **conforming** cell at PCR rate was **not** received at TAT3 or within TAT3's CDVT the burst containing Cells 1 and 3 is over so the counter is reset to 0 to prepare for the next burst. Remember, from the perspective of providing a "gap" between bursts, an SCR or MBS nonconforming cell is just like an Idle Cell.
  - **Cell Count = 2**: Since the cell was not Burst conforming the Cell Count is **not** incremented.
- Cells 10, 11, and 12 are Idle Cells.
- Cell 13 is a CLP(0) data cell.
  - **Burst Count = 1**: Since there was a PCR TAT *gap* after the last conforming data cell (TAT4) this is the first cell of a new burst. The count is less than or equal to the MBS (2) so it is Burst conforming.
  - **Cell Count = 3**: Since the cell was Burst conforming we now see if the result of incrementing the Cell Count would be less than or equal to the SCR cps (6). It is, so the cell is conforming to the SCR and we increment the counter.
- Cell 14 is a CLP(0) data cell deemed nonconforming by the PCR/CDVT test and discarded so its time slot in the stream is treated the same as an Idle Cell.
- Cells 15 and 16 are Idle Cells.
- Cell 17 is a CLP(0) data cell.
  - **Burst Count = 2**: Since the cell was received at **or before** the TAT from the previous data cell it's considered part of the same burst. Since the result of incrementing the

burst counter would be less than or equal to the MBS (2) it is Burst conforming so the counter is incremented.
    - **Cell Count = 4**: Since the cell was Burst conforming we now see if the result of incrementing the Cell Count would be less than or equal to the SCR cps (4). It is, so the cell is conforming to the SCR and we increment the counter.
- Cells 18 and 19 are Idle Cells.
- Cell 20 is an Idle Cell. Since it represents the last slot for the one second period:
    - **Burst Count = 0**: The Burst Counter is reset to prepare for the next burst.
    - **Cell Count = 0:** The Cell Counter is reset.

The good news at this point is that you should now have a basic understanding of VBR traffic policing. The bad news is that there are still two more VBR algorithms to cover. There really isn't much difference from what we have already covered so only one example will be provided for each.

PCR/CDVT (0+1), SCR/MBS (0) - No Tagging

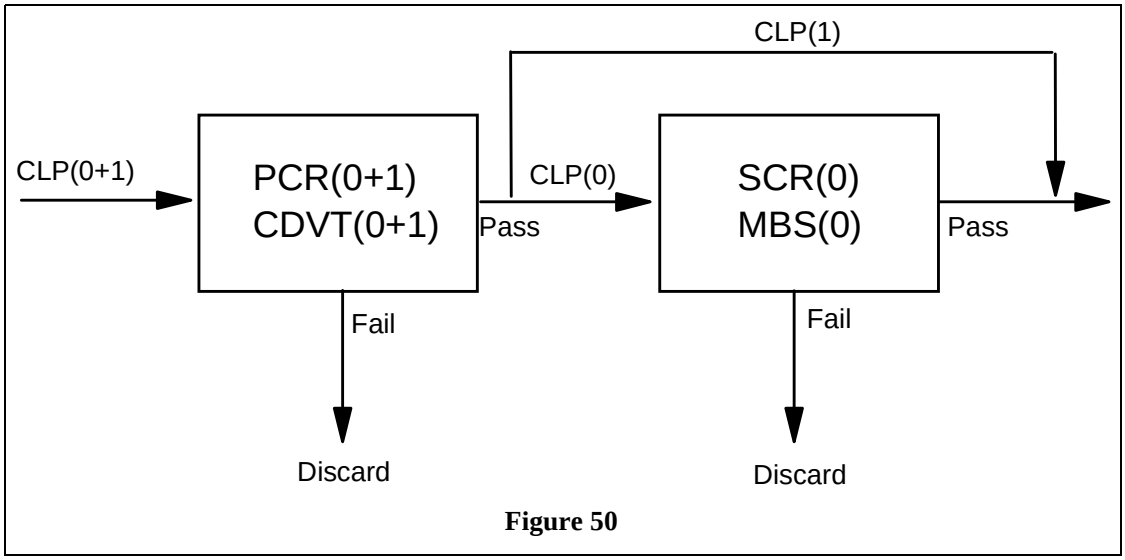

Figure 50

In this algorithm, only the cells that passed the initial PCR/CDVT test **and** have the CLP bit set to 0 (indicating conformance) are subjected to the SCR/MBS test. Cells that don't conform to the second test are discarded.

Since the basic operation is the same as the SCR/MBS (0+1) algorithm just covered we'll start right into the example.

SCR(0)/MBS(0), No Tagging Policing: Example 1

Determine which of the following data cells are conforming and which are nonconforming using the indicated traffic parameters. Remember:

- CLP=1 cells are indicated by underlined cell numbers.
- The CLP=1 cells are subjected to the PCR/CDVT test but are ignored by the SCR/MBS test.

Please take note of the unusual, though valid, PCR value.

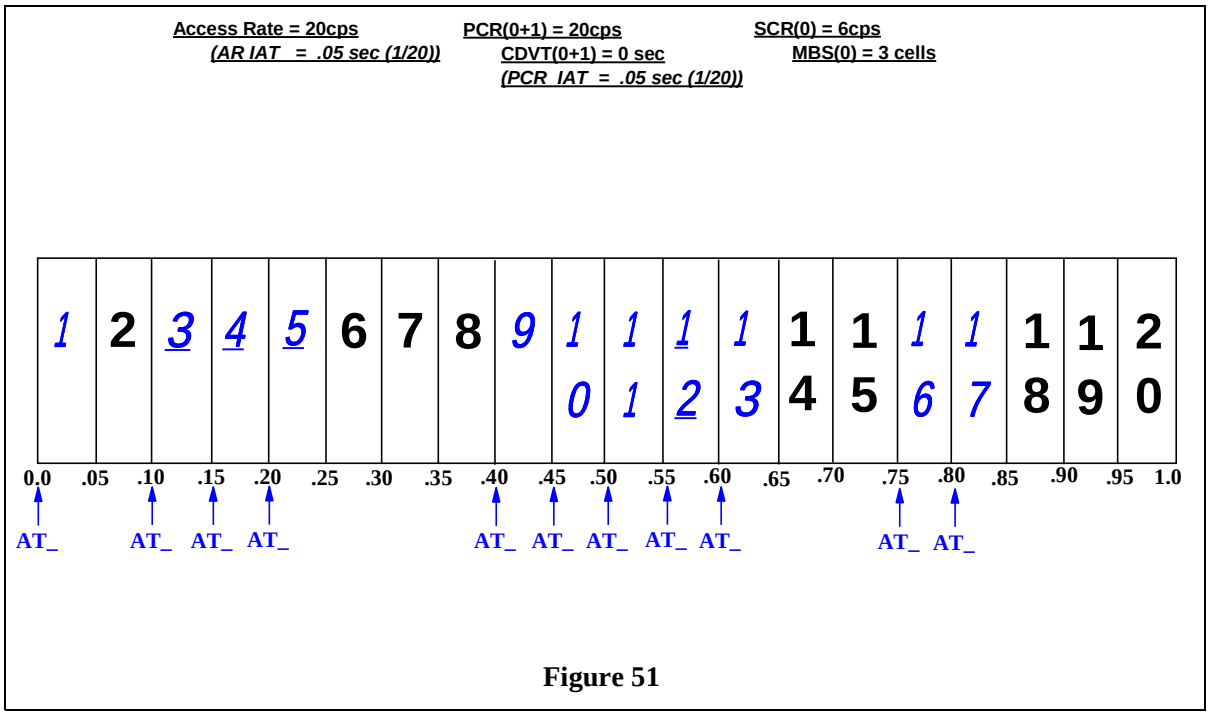

Figure 51

The PCR/CDVT policing should result in the following:

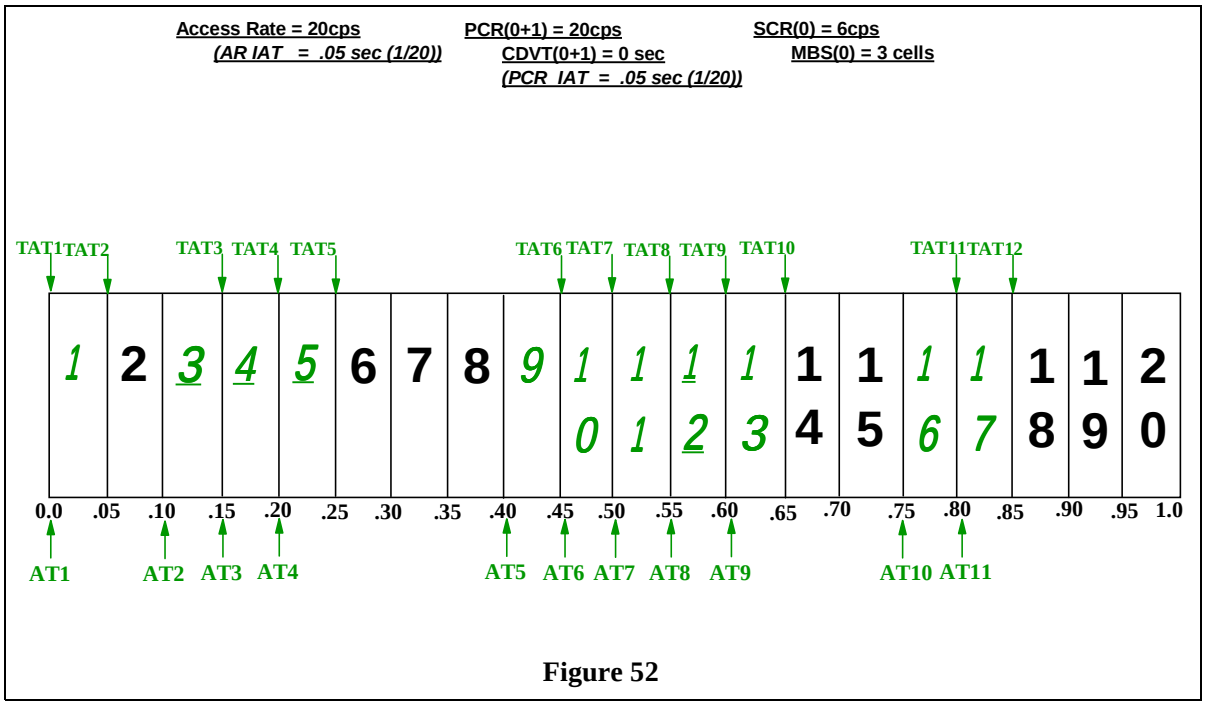

Figure 52

All the cells were conforming. PCR cps was set to the Access Rate of the line (remember PCR can be less than **or** equal to the AR) so it was impossible for a cell to be nonconforming in the First Stage. Notice how the TATs all point to the next slot. I had to do this to be able to fit in enough conforming cells for the example (one of the few disadvantages of the ***MATT*** interface).

The SCR/MBS policing should result in the following:

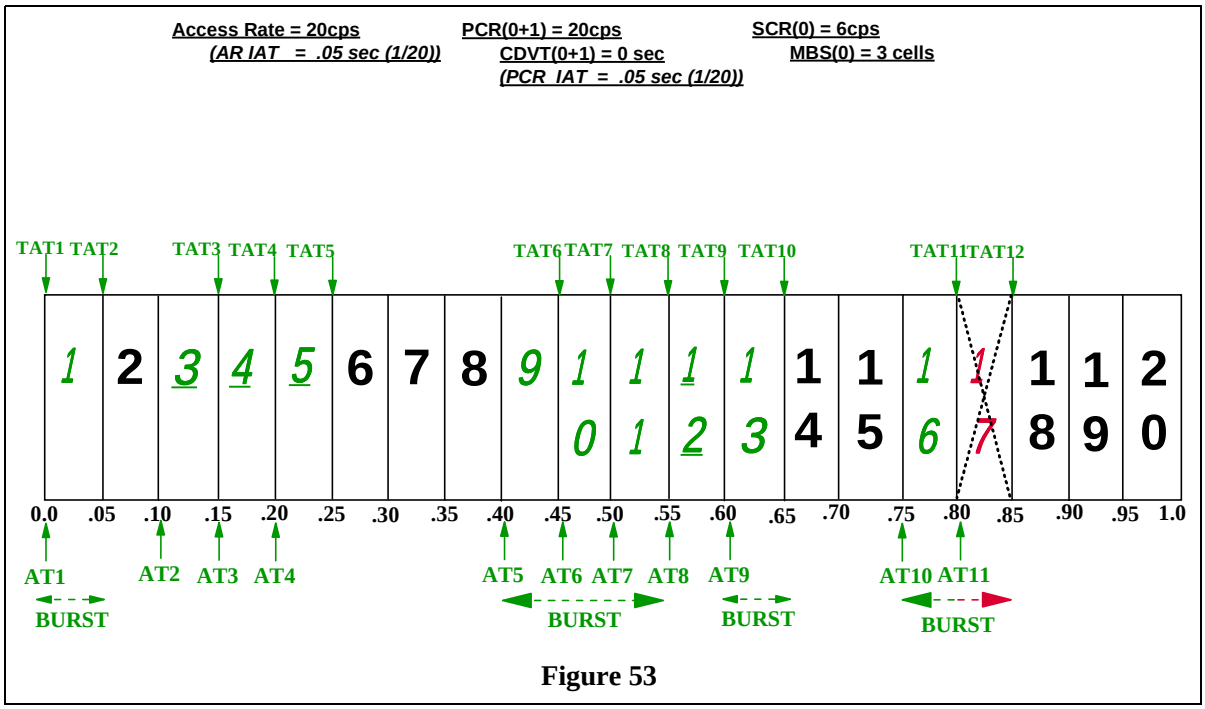

Figure 53

- Cell 1 is a CLP(0) data cell and was received at time .00sec. It is the first data cell as well as the first cell of a burst so:
  - **Burst Count = 1**: This is less than or equal to the MBS (3) so it is Burst conforming.
  - **Cell Count = 1**: This is less than or equal to the SCR cps (6) so it is SCR conforming.
- Cell 2 is an Idle Cell but it's where the next data cell at PCR rate would have been (TAT2). This means the burst containing only Cell 1 is over.
  - **Burst Count = 0**: The Burst Counter is reset to prepare for the next burst.
  - **Cell Count = 1**: The cell count is not reset since it keeps track of the total number of cells that can be sent in one second.
- Cells 3, 4, and 5 are tagged (CLP=1) data cells so they are ignored. From the perspective of SCR/MBS conformance they are treated the same as Idle Cells.
- Cell 6, 7, and 8 are Idle Cells.
- Cell 9 is a CLP(0) data cell.
  - **Burst Count = 1**: Since there was a PCR TAT *gap* after the last conforming data cell (TAT2) this is the first cell of a new burst. The count is less than or equal to the MBS (3) so it is Burst conforming.
  - **Cell Count = 2**: Since the cell was Burst conforming we now see if the result of incrementing the Cell Count would be less than or equal to the SCR cps (6). It is, so the cell is conforming to the SCR and we increment the counter.
- Cell 10 is a CLP(0) data cell.
  - **Burst Count = 2**: Since the cell was received at or before the TAT from the previous data cell it's considered part of the same burst. Since the result of incrementing the burst counter would be less than or equal to the MBS (3) it is Burst conforming so the counter is incremented.

    - **Cell Count = 3**: Since the cell was Burst conforming we now see if the result of incrementing the Cell Count would be less than or equal to the SCR cps (6). It is, so the cell is conforming to the SCR and we increment the counter.
- Cell 11 is a CLP(0) data cell.
    - **Burst Count = 3**: Since the cell was received at or before the TAT from the previous data cell it's considered part of the same burst. Since the result of incrementing the burst counter would be less than or equal to the MBS (3) it is Burst conforming so the counter is incremented.
    - **Cell Count = 4**: Since the cell was Burst conforming we now see if the result of incrementing the Cell Count would be less than or equal to the SCR cps (6). It is, so the cell is conforming to the SCR and we increment the counter.
- Cell 12 is a tagged data cell (CLP=1) so it is **not included as part of the burst**. From the perspective of SCR/MBS conformance it is treated the same as an Idle Cell. It's located where the next conforming data cell at PCR rate would have been (TAT8) so the burst containing Cells 9, 10, and 11 is over.
    - **Burst Count = 0**: The Burst Counter is reset to prepare for the next burst.
    - **Cell Count = 4**: The cell count is not reset since it keeps track of the total number of cells that can be sent in one second.
- Cell 13 is a CLP(0) data cell.
    - **Burst Count = 1**: Since there was a PCR TAT *gap* after the last conforming data cell (TAT8) this is the first cell of a new burst. The count is less than or equal to the MBS (3) so it is Burst conforming.
    - **Cell Count = 5**: Since the cell was Burst conforming we now see if the result of incrementing the Cell Count would be less than or equal to the SCR cps (6). It is, so the cell is conforming to the SCR and we increment the counter.
- Cell 14 is an Idle Cell but it's where the next data cell at PCR rate would have been (TAT10). This means the burst containing only Cell 13 is over.
    - **Burst Count = 0**: The Burst Counter is reset to prepare for the next burst.
    - **Cell Count = 5**: The cell count is not reset since it keeps track of the total number of cells that can be sent in one second.
- Cell 15 is an Idle Cell.
- Cell 16 is a CLP(0) data cell.
    - **Burst Count = 1**: Since there was a PCR TAT *gap* after the last conforming data cell (TAT10) this is the first cell of a new burst. The count is less than or equal to the MBS (3) so it is Burst conforming.
    - **Cell Count = 6**: Since the cell was Burst conforming we now see if the result of incrementing the Cell Count would be less than or equal to the SCR cps (6). It is, so the cell is conforming to the SCR and we increment the counter.
- Cell 17 is a CLP(0) data cell.
    - **Burst Count = 2**: Since the cell was received at or before the TAT from the previous data cell it's considered part of the same burst. Since the result of incrementing the burst counter would be less than or equal to the MBS (3) it is Burst conforming so the counter is incremented.
    - **Cell Count = 6**: Since the cell was Burst conforming we now see if the result of incrementing the Cell Count would be less than or equal to the SCR cps (6). **The**

result would be greater than the SCR cps so the Cell Count stays at 6. Since this also makes the cell nonconforming, it will be discarded.

    - **Burst Count = 0**: Since a **conforming** cell at PCR rate was **not** received at the next TAT (TAT11- Cell 17's time) the burst containing Cell 16 is over so the Burst counter is reset to 0 to prepare for the next burst.
- Cells 18 and 19 are Idle Cells.
- Cell 20 is an Idle Cell. Since it represents the last slot for the one second period:
    - **Burst Count = 0**: The Burst Counter is reset to prepare for the next burst.
    - **Cell Count = 0:** The Cell Counter is reset.

So after all that only one cell ended up being discarded. Hardly seems worth the effort.

There's now just one more algorithm to cover.

PCR/CDVT (0+1), SCR/MBS (0) - Tagging

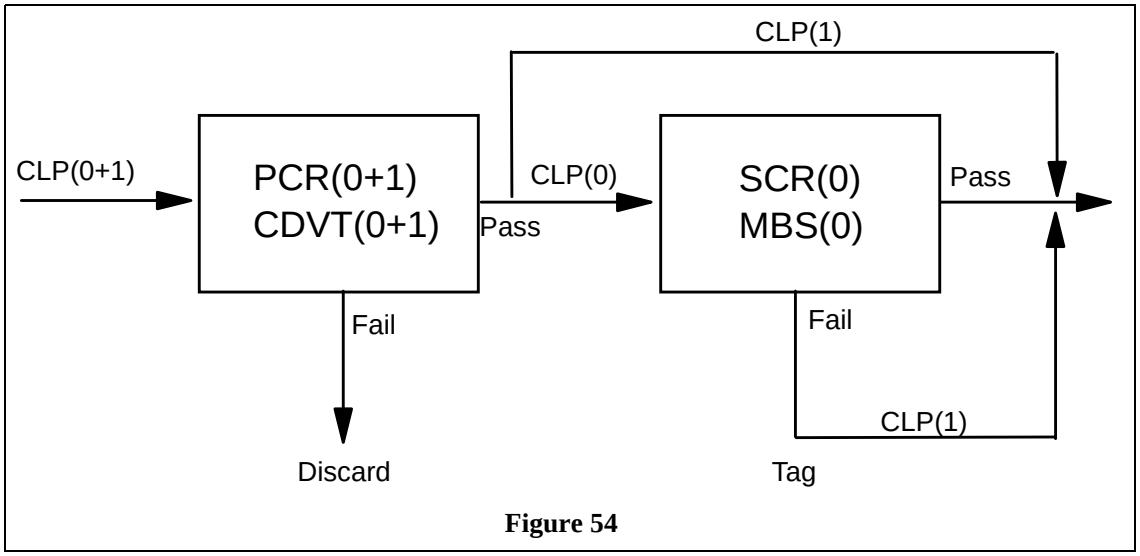

Figure 54

This is exactly the same as the previous algorithm except that cells which don't conform to SCR/MBS are tagged and passed into the network instead of being discarded.

SCR(0)/MBS(0), With Tagging Policing: Example 1

Determine which of the following data cells are conforming and which are nonconforming using the indicated traffic parameters. Since these values are identical to the previous example's, I really hope most of you already know what will be different.

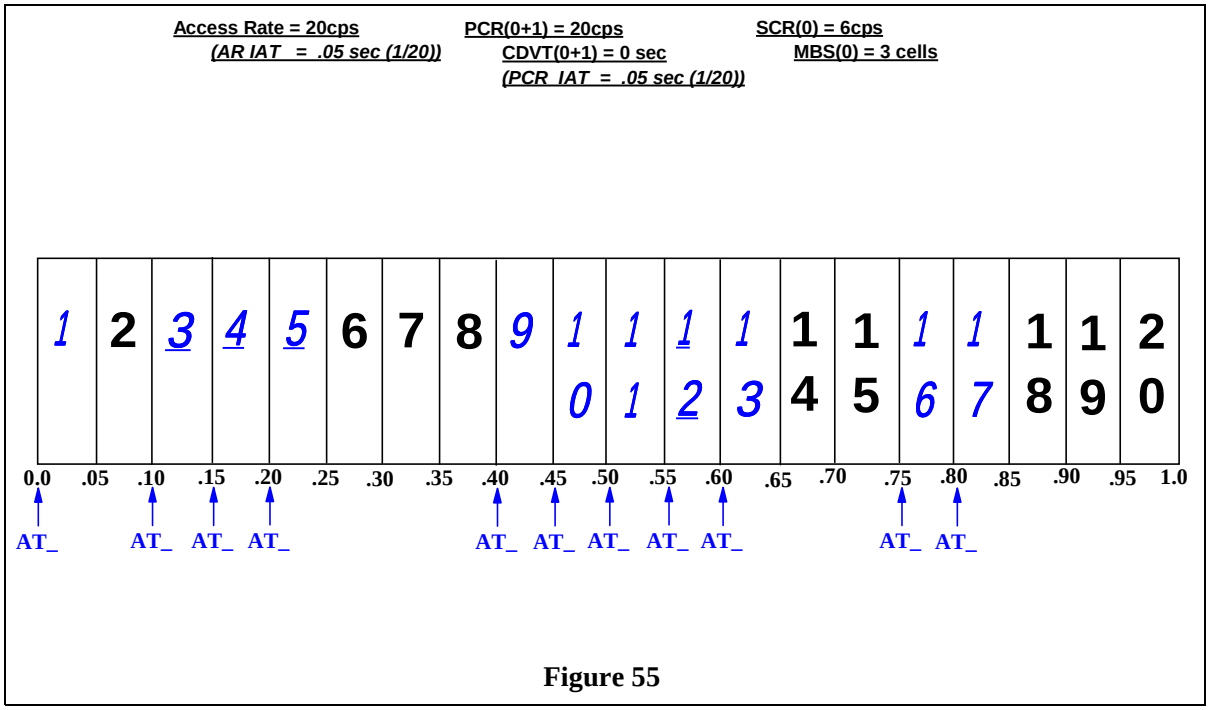

Figure 55

The PCR/CDVT policing should have the same result as shown in Figure 52 (page 76) since this part of the algorithm is the same. What about the SCR/MBS test?

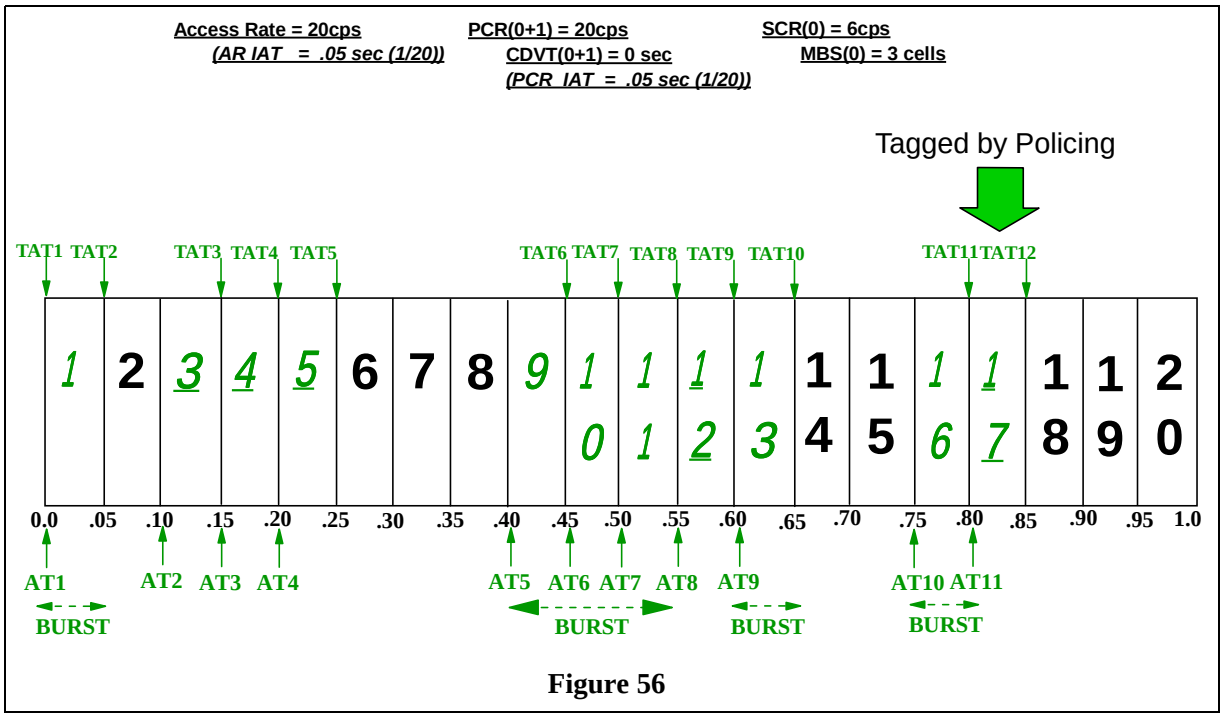

Figure 56

The only difference was that Cell 17 was tagged and passed on instead of being discarded. Since it was a nonconforming cell all the resetting of counters and ending of bursts in the cell by cell description would be the same as when the cell had been discarded.

Summary

Congratulations to all (any?) of you who actually read the entire paper and tried to work through the examples before looking at the answers. Why do I feel like I'm speaking in front of an empty auditorium?

In most cases, the only way of determining accurate and efficient traffic descriptors is to hook up an ATM analyzer to see the actual characteristics of the traffic you will be policing. If you are not able to do this, hopefully what you've learned in this paper will allow you to make some educated guesses.

If you are experiencing excessive cell discards, your options will depend on the type of policing algorithm you are using.

- PCR/CDVT Algorithms
  - The PCR value may just be too small for the amount of traffic. Try raising it until the discards stop.
  - If you know/think your PCR value should be big enough then the problem must be CDVT related. There could be two situations you could address by increasing the CDVT (if you're allowed to modify the value):
    - Cell Clumping is occurring. Small increases in the CDVT value should be tried.
    - The bursts are not being shaped by the transmitting device or you are multiplexing several VCCs over a VPC you are policing. Increasing the CDVT value enough to cover the maximum burst size (like we did in **One Stage PCR/CDVT Policing: Example 8** on page 29) would help. Coming up with a good value, especially in a VPC scenario, can usually only be done via trial and error.
- SCR/MBS Algorithms
  - Everything said above for the PCR/CDVT algorithms can also have an impact here since that testing is always done first.
  - Cells can be discarded in the Second Stage either because your bursts are too long or the number of burst per second ends up exceeded the SCR value.
    - Knowing the characteristics of the applications generating the traffic is necessary to come up with an accurate value for MBS. If you don't have that information start with a value to cover the absolutely largest frame that you can imagine and start lowering it until cell discards stop occurring.
    - Coming up with a good MBS value is easy compared to SCR. Although you can guess at a maximum burst size it's much more difficult to come up with a good guess on how many bursts will occur per second. Once you're comfortable with your MBS value you then just have to start trying different SCR values.

If you're able to get the amount of discards down to a *reasonable* level doing one of the above you should then consider using one of the algorithms that has a tagging option. This way you won't be using more bandwidth than you need on average but still allow the cells to get through when the traffic rate increases from time to time.

My goal in writing this paper was to provide a practical explanation of ATM Policing via simplified algorithms and basic examples that non-mathematicians could understand. Though the ***MATT*** interface has not yet gained widespread industry acceptance you can hopefully use what you've learned to make at least informed guesses as to why you may be experiencing excessive cell discards and then take appropriate actions to try to eliminate them. You should also now be in a better position to try to understand the more formal documentation on the topic. You know, the stuff with all the $\sum$ and α symbols.

As mentioned in the *Preface*, based on the feedback I receive on this paper I would like to produce a companion document titled *Introduction to ATM Traffic Descriptors.* The goal would be to provide practical guidance on how to select appropriate Policing algorithms and descriptor values for different types of application traffic and scenarios. Let me know what you think (**trzyna@us.ibm.com**).

Appendix A: Recommended Reading

- de Prycker, Martin, *Asynchronous Transfer Mode: Solution for Broadband ISDN*, Prentice Hall, 1995
- Giroux, Natalie and Ganti, Sudhakar, *Quality of Service in ATM Networks*, Prentice Hall, 1999
- Goralski, Walter J., *Introduction to ATM Networking*, McGraw-Hill, Inc., 1995

Appendix B: Glossary

- **Access Rate (AR):** The maximum data rate physically possible on an interface
- **Access Rate IAT (AR IAT):** IAT of cells at Access Rate. AR IAT is calculated by: 1/AR.
- **Call Admission Control (*CAC*):** The process of accepting or rejecting the start of a connection.
- **Cell Rate Decoupling**: Process of inserting Idle Cells into the ATM cell stream during transmission.
- **Cell Delay Variation Tolerance (*CDVT*)**: A time value (multiples of the AR IAT) which specifies how much sooner a cell can arrive than what would be expected by the PCR IAT and still be considered conforming.
- **Cell Clumping:** Condition in which cells are received sooner than the PCR rate due to delays in the network.
- **Cell Loss Priority (*CLP*)**: Status of a bit in the ATM cell header which identifies the priority of the cell for discard if there is congestion within the ATM network. Cells with CLP=1 are supposed to be discarded within the network before cells that have CLP=0.
- **Idle Cell**: Type of cell inserted into the ATM Cell stream by the transmitting device at the next scheduled cell time when there are no data cells to transmit.
- **Inter-Arrival Time (*IAT*)**: The time between start of the receipt of one cell and the start of the receipt of the next. IAT is calculated by: 1/rate.
- **Maximum Burst Size (*MBS*)**: The maximum number of cells the connection can send back to back at Peak Cell Rate. The specified value must be less than or equal to the SCR.
- **PCR IAT**: IAT of cells at the PCR rate. PCR IAT is calculated by: 1/PCR.
- **Peak Cell Rate (*PCR*)**: The maximum rate, in cells per second, at which a virtual circuit can transmit. The specified value must be less than or equal to the Access Rate.
- **Policing:** Action taken by the ATM switch to ensure incoming traffic does not exceed the rate defined by the traffic descriptors which were agreed on between the switch and the traffic source when the virtual circuit was established.
- **Sustainable Cell Rate (*SCR*)**: The average rate, in cells per second, at which a virtual circuit can transmit. The specified value must be less than or equal to the PCR.
- **Theoretical Arrival Time *(TAT)***: This indicates the soonest the next cell can arrive and still be considered conforming. Its value is calculated by adding the *PCR IAT* to the *Arrival Time (AT)* of the last conforming cell: *AT + IAT = TAT*.