

DON'T DO THAT! (IPTV VERSION)

WRITTEN BY JIM FARMER, CTO & KEVIN
BOURG, SR. TECHNICAL DIRECTOR
ORIGINALLY PUBLISHED IN FTTH PRISM

When I (Jim) was a kid, it seemed as if all I heard was, “Don’t do that!” But then, I was a pretty bad kid from all accounts. Then I got married, and all I heard was “Don’t do that!” My parents finally got rid of me, but she’s been stuck with me for a lot longer than they were. Anyway, that phrase has resonated with me, so when Kevin and I were talking about papers for the 2008 FTTH Conference, I urged (nay, browbeat) him to do something along the lines of how not to do IPTV. As we dove back into the experiences of people we’d worked with, we found some interesting examples of things you can do wrong with IPTV, which can cause you big headaches. We wrote the paper and Kevin presented it at the Conference in Nashville. Go back and read it.

There are a number of ways to mess up any business or technology, and IPTV is no exception. There are several popular ways to get into trouble. One way is to design your network so that you don’t properly protect your content. When you let subscribers and maybe even non-subscribers rip you off, not only do you lose money, but the folks from whom you buy content, especially premium content, get really mad.

DON’T BOTHER SCRAMBLING

This one guy figured that, since we encrypt content between the OLT and the ONT, he didn’t have to spend money on video scrambling (TV folks use the word scrambling rather than encryption for historical reasons, but in this context, they are the same thing). He programmed his set tops to request only the content each subscriber had paid for. What he forgot was that anyone with more computer savvy than I have (and that includes most teens) can use a computer to request content that the set top was not allowed to request. So since he had not scrambled the program, the system delivered it to the ONT then to the computer, just as it was supposed to do.

Of course, once you get the video content onto a computer, in the clear, what’s to stop you from sending it to all your buddies, or even posting it to the web? Sure beats some of the stuff I see on social networking sites. And that makes the content owners (the studios or whoever) really, really nervous. They see what happened to music sales when people started distributing music on the Internet (whether or not that was the only reason for the music industry’s problems is open for debate, of course), and they see themselves as next. His first solution, once he understood the problem, was to use ACLs, or access control lists, to limit who got what. ACLs are really neat things you put on data switches, to control where information goes. You tell the switch to send or not to send information from this source (which can be a TV channel) to that subscriber. ACLs are useful in deciding who gets access to what kind of info, and are primarily used in combating denial of service attacks. Entering the ACLs for every subscriber has to be done manually, while adding new subscribers and keeping up with subscribers who keep changing their mind about what service they want. All this turned out to be a nightmare: an error-prone, time-consuming nightmare. The answer was to let the FTTH system do what it was designed to do, namely deliver an incredible amount of data to the subscriber at a great cost model. Then let the set top system do what it was designed to do, prevent programming from reaching subscribers who have not paid for it, while delivering it conveniently to subscribers who have paid for it. And that takes scrambling.

DON’T WORRY ABOUT YOUR VLAN CONFIGURATION

Before we get to the next problem, let’s get in some background on the subject of virtual local area networks, or VLANs. You know what a local area network is: that’s simply a group of computers connected together on a network. Any computer can access any other computer on the network, within of course, the limits of authorizations set up on the accessed computer. Well, in data switches you can set up a virtual local area network that acts the same way, except that part of the network uses the same communications path (e.g., an FTTH system) as does other data. The VLAN function keeps the data separate, so that the same functions happen as on a simple local area network. This is a good way, for example, to prevent TV signals from getting to the wrong people: you put TV in a VLAN and only extend that VLAN to ONTs of people who have paid for

TV. The VLAN is handled using special tags in Ethernet frames, and these tags can be added or deleted (either adding or stripping the VLAN) at various points along the route, up to and including at the ONT.

In a variation on the first theme, another fellow had built a simple network in which he had put scrambled programming, and programming being sent to the scramblers, on the same VLAN. Of course this VLAN had to be open to subscribers, in order for them to get programming they had paid for. The fellow also assumed that the set top was protecting him. But he ran into the same problem, with folks accessing programs being sent to his scramblers. This is sort of like the cable TV systems which used to have such bad record-keeping that they didn't disconnect subscribers who dropped the service. If you don't want people to get something for free, don't give it to them. Of course you could ask them nicely to just not watch programs they didn't pay for, but you may find that this doesn't work too well.

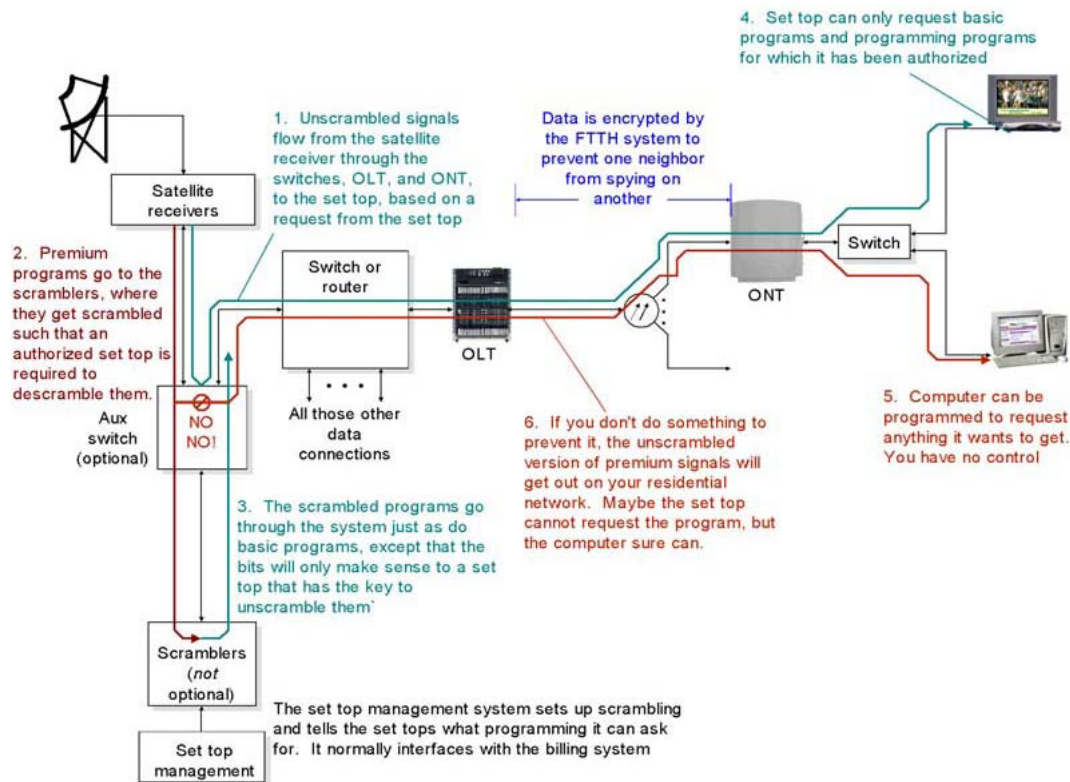


Figure 1. Letting unscrambled signals into the wild

Figure 1 illustrates these first two cases at a high level. Satellite receivers and associated equipment receive programming and put it into proper IPTV format. Some of the programs are basic programs intended for all subscribers, so they can be released “into the wild” of the FTTH network without scrambling (though many folks scramble them anyway). You will have to make sure that non-TV subscribers can't get the programs on their computers, but you can do that by making sure the VLAN that contains the TV programs doesn't go to ONTs of non-TV subscribers. Or you can scramble everything, which is why many folks do it.

The premium programs, which must be scrambled by the set top scramblers, are routed to the scramblers shown in the lower left. Programs going to the scramblers must be isolated from the programs going to subscribers. This can be done by physically isolating paths, putting them on their own VLAN that doesn't go to subscriber, or by using a layer three boundary. The problem in one of our examples above was that the

programs being sent to scramblers were in the same VLAN as those going to subscribers. So there was effectively a “shortcut,” shown in the auxiliary switch (used just for video switching), that let a computer request any program regardless of whether it is going to the scramblers or not. How did subscribers know how to program the requests? One way is to try all possible IGMP join requests (the method used by IPTV set tops to “tune” a channel), but another is to snoop data at the home of a legitimate subscriber using a cheap data switch and a freeware data monitoring program.

The first attempt the operator made to put his finger in the dam was to use access control lists, ACLs, on the switch. ACLs are a way of permitting or denying data to a client on the network. This way, he could deny programming to any ONT that had not paid for that programming. One problem was that there is no logical connection between the billing system and the ACL list in the switch, so every time a subscriber made a change in service, the ACL had to be programmed differently. This process is very error-prone, so after a while, the guy who decided to do this was spending way too much time chasing errors in the ACLs. Furthermore, as the system grew and the ACLs got more complex (with multiple tiers of premium programming to protect), the ACLs got unwieldy. And think what would have happened had our hapless friend tried to offer video-on-demand (VoD), in which the subscriber chooses a movie and expects to start getting it within a few seconds.

Had he gotten that far, our operator would have run into severe DRM issues, since he was letting subscribers get high-value content in unscrambled form, from whence they could do with it as they pleased. Had his program suppliers found out what he was doing, he would likely have lost his access to TV. The answer is to scramble programs and let the set top do what it was designed to do.

Are set tops perfect security? Nope, not by a long shot. But the biggest holes are kind of plugged, and Hollywood has accepted certain safeguards built into set tops, as a reasonable compromise between security and subscriber convenience. This subject is often called digital rights management, or DRM. It’s a can of worms, but maybe Dave will let us get into that in a future edition of the FTTH Prism. [Ed. Note: Sounds like a winner, Jim.] While there is no universally-accepted definition of DRM, it is the whole process of controlling who gets what content and what they can do with it once they get it. It includes how the picture and sound are scrambled, how the subscriber gains authorization to the program, and how to control what he does with it after he gets it. For instance, the program owner may insist that the subscriber can only display the signal on a TV at the time he gets it. Or the subscriber may be allowed to record it once, or he may be allowed to record it a limited number of times. He may or may not be allowed to move the material onto a computer or a portable viewing device (another process more complex than it seems on the surface). There are technical teeth behind each of these authorizations – Hollywood has a bias against nicely asking folks to not do something they have not paid to do – they want to take a bite out of anyone who tries to go beyond what the owner wants done with the content.

CONFIGURING A SWITCH? WHAT’S THAT?

We’ve seen other ways to get into trouble, too. For instance, when you design your network, you have to make sure IPTV packets, the slivers of information the set top uses to generate the picture, get to the set top when the set top needs them. TV (and voice) is most unforgiving of packets that get delayed so that they don’t arrive in time. When the set top needs the information in the packet, it needs it now, not a fraction of a second later. If it gets it late, either the picture pixilates or it freezes. What is pixilation? You’ll know it when you see it, and you won’t like it. Makes subscribers very angry.

Every system has points where the data can be constricted, and you better manage them well. Any time you switch video and other data, you need to make sure you have your quality of service (QoS) parameters set right, as well as seeing to it that you have enough bandwidth. This has gotten some operators in a lot of hot water, and trouble-shooting the problem is tough. Give video too little of the bandwidth resource through the point of constriction and it can’t all get through. Give video too much of the bandwidth, and other data starts

snarling traffic by not getting through, causing retransmissions: the effect is to flood the constriction point with more data, not less – if you're confused, don't worry, because we have an example below. Sort of like trying to cure our too-crowded expressways here in Atlanta by closing the surface streets to force more cars on the expressway. Is there a solution? Sure, but you need the expertise of someone who understands how to manage a mix of traffic. As with so much in life, it's easy (well, fairly easy, or at least easier) when you understand it.

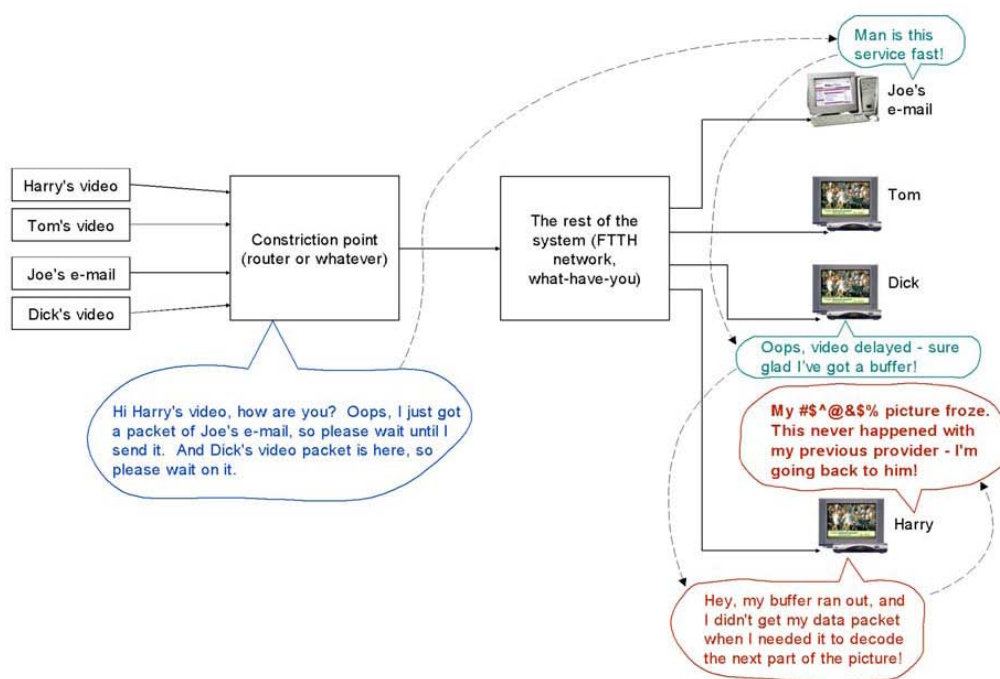


Figure 2. Constriction point where all packets are treated equally

Figure 2 illustrates a constriction point at which all packets are treated equally – sounds good on paper, but we've seen operators get in a lot of trouble this way. The constriction point might be a big router, or a switch, which handles massive amounts of data of all types from a number of sources, as in this example. What we have in the simple example, is one customer, Joe, who is receiving his email. We also have three customers, Tom, Dick, and Harry, who are watching different IPTV programs. All data are feeding through one big switch, which is the constriction point in this case. That switch feeds an FTTH network and anything else that may be in the common path of the data (such as other constriction points).

Follow the dashed lines to see what's going on. At the moment of our example, the router-constriction point has just gotten a packet of video for Harry. But he also gets a packet of email for Joe, and since he is programmed to treat all packets equally, he sends Joe's email packet first and makes Harry's video packet wait. He also has Dick's video packet, which he sends right after Joe's email packet. Then he finally gets around to sending Harry's video packet. So what experience do our subscriber get? Well, Joe's email came fast, so he's thinking about how good a provider you are.

Dick's video packet was delayed some, but his set top, like all IPTV set tops, has buffer memory. That is, it takes in packets and holds them for a short time before passing them on to the decoder. Why? Because we

know that in any packet network there will be packet jitter, meaning that the packets will not arrive exactly at the time they are needed: every time a packet arrives at a switch, there is the possibility that it will be delayed while another packet is sent. So set top designers put in a buffer to compensate. How big (i.e., how much buffer time is provided) is a matter of concern to the manufacturer. Put in too little buffer and you might run out of packets as in Harry's case that we'll see shortly. Put in too much buffer and you spend money on a very, very cost-competitive product, and you will lengthen channel change time. We know how much attention channel change time is getting in the industry. But anyway, thanks to the buffer in Dick's set top, the delay in his video didn't cause a picture or sound problem, so Dick is happy.

Harry, on the other hand, is not in such good shape. His packet got delayed the most by the router at the constriction point, so it is the latest to get to its intended destination, Harry's set top. As a result of the delay, Harry's decoder didn't have the data it needed when it needed it. Harry's picture froze. Depending on the decoder and just what data it misses, the picture may freeze, or it may pixelate. We can't print what Harry said, but we get the idea that he is not happy with us, and he is thinking of changing back to the competition because of his video problems.

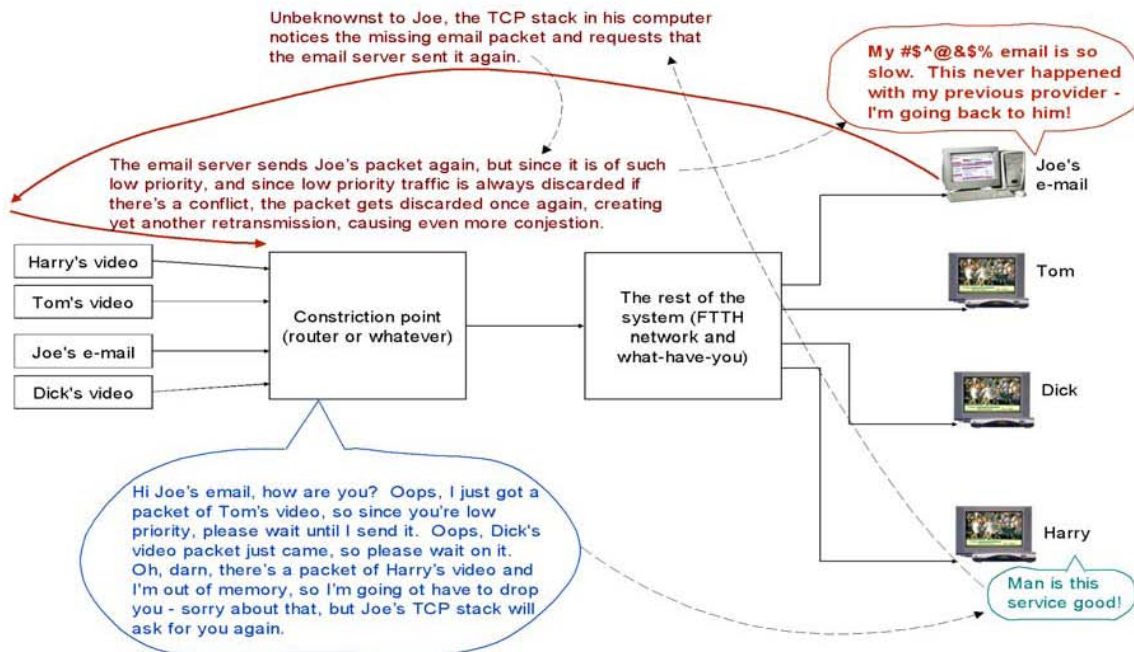


Figure 3. Constriction point with QoS that favors video with no allowance for other packets

Figure 3 illustrates what can happen when you go too far the other way. In this case, the constriction point is configured to let video packets through at highest priority, and if the constriction point gets into trouble, tough for any other type of packets (i.e., it is programmed for early discard with no weighting for fairness). In this case, the constriction point gets a packet containing email for Joe. But it also gets a video packet for each of Tom, Dick, and Harry. It sends the video packets one after the other, but in the process, it gets into trouble. When a packet is delayed, the router must have enough memory to hold that packet until it is able to send it on. Suppose that, due to momentary congestion, it runs out of memory to store everything. So Joe's email packet gets dropped (euphemistically, discarded).

Meanwhile, Harry's video is getting through fine, so he's a happy guy. Joe, on the other hand, is about to get mad at us. Let's see what happens to his email packet. You don't see it, but email and many other data packets are transferred using a protocol called TCP, transmission control protocol. When someone talks about TCP/IP, this is part of what they mean. TCP makes sure that all packets arrive at their destination, and if they arrive out of order (which can and does happen), they put the order back correctly before handing the message to the email program. We call the TCP program a stack. When the TCP stack realizes that there is packet missing from Joe's email, it requests the email server (not shown) to resend the packet. Dutifully, the email server does so. But when the resent packet hits the constriction point, if it is still constricted, the packet could be dropped once again. A few seconds later, Joe's email stack again figures out that the packet is not going to come, so it requests yet another transmission. Thus, we have jammed up our constriction port with three packets in order to get one through. We have added to the burdens of the constricted point, rather than alleviated them. And Joe is one mad subscriber because he isn't getting his stuff.

AT LAST, THE CONCLUSION

IPTV is a good tool for use in FTTH networks when you need to deliver one-to-one video, and it can work for traditional broadcast programming, too. But you have got to know what you're doing. If you really (be honest now) have the expertise in-house, go for it. But few IT guys understand the needs of IPTV. If in doubt about your expertise, find someone who knows how to configure your network for efficient IPTV, and let him or her do their thing. You may have a vendor who knows their stuff, but if you don't, get a third party involved. Otherwise, you may find yourself saving money that turns out to cost you a lot – as we called it in the Conference paper, the most expensive money you ever saved.

**For more information
visit www.enablence.com**

©2010 Enablence Technologies Inc. The information presented is subject to change without notice. Enablence Technologies Inc. assumes no responsibility for changes or inaccuracies contained herein. Copyright © 2010 Enablence Technologies Inc. All rights reserved.