International Journal of Advanced Information Science and Technology (IJAIST) ISSN: 2319:268
Vol.2, No.5, May 2013 DOI:10.15693/ijaist/2013.v2i5.33-37

Performance Analysis of Traffic Flow in Wide area Network

Ashima
Student/Department of CSE
Amity University, Noida, India

Sanjeev Thakur
Assistant Professor/Department of CSE
Amity University, Noida

Abstract— **Today's Internet only provides Best Effort Service. Traffic is processed as quickly as possible, but there is no guarantee of timelines or actual delivery. Different types of Traffic Management systems are used for internet services. Queuing is one of the very vital mechanisms in traffic management system. Congestion management entails the creation of queues, assignment of packets to those queues based on the classification of the packet, and scheduling of the packets in a queue for transmission. The queue management algorithm, which is applied to a router, plays an important role in providing Quality of Service (QoS).There are different queue management algorithm like First in First out (FIFO), Priority queue (PQ) and Weight Fair Queuing (WFQ). In this research paper , a comparison was carried out between two different queuing algorithms class based weighted fair queuing (CBWFQ) and low latency queuing (LLQ).**

Index terms - Bandwidth, Congestion Control, Traffic Management, Queuing, Quality of Service.

I. INTRODUCTION

Wireless Recently, the exponential growth of the Internet and the use of new services such as e-business, voice over IP (VoIP) and multimedia applications has been risen the need of supporting Quality of Service (QoS) requirements. The queue management algorithm, which is applied to a router, plays an important role in all of QoS measurements (include delay, jitter, bandwidth and packet loss). Queuing management algorithm is responsible for accepting the arriving packets or not accepting them and consequently it directly affects the packet loss quantity parameter. The queue management algorithm also should ideally keep the queue occupation level as low as possible, to ensure low delay. However, to ensure maximum utilization of the outgoing link, the queue should never be empty. Furthermore, maintaining queue stability is important as some applications are sensitive to jitter. At the source, where flow control algorithm vary the rate at which the source sends packets. Flow control algorithms are designed to ensure the presence of free buffers at the destination host. Also congestion can be controlled at gateways through routing and queuing algorithms [1]. Queuing algorithms, which control the order in which packets are sent and the usage of gateways buffer space, do not affect congestion directly but determine the way in which packets from different sources interact with each other, which in turn affects the collective behavior of flow control algorithms.

II. QUALITY OF SERVICE IN NETWORK

The QoS is measured according ITU recommendations based on different parameters like (delay, jitter, and packet loss), these parameters can be changed and controlled within the acceptable range to improved QoS. Factors affecting QoS are briefly described in the following sections [2]:

A. Latency

As a delay sensitive application, voice cannot tolerate too much delay. Latency is the average time it takes for a packet to travel from its source to its destination. A person whose speaking into the phone called the source and the destination is the listener at the other end. This is one-way latency [3]. Ideally, must keeping on the delay as low as possible but if there is too much traffic on the line (congestion), or if a voice packet gets stuck behind a bunch of data packets (such as an email attachment), the voice packet will be delayed to the point that the quality of the call is compromised [4]. The Maximum amount of latency that a voice call can tolerate one way is 150 Milliseconds (0.15 sec) but is preferred be 100 Milliseconds (0.10 sec) [5]. Formula shows the calculation of Delay where the Average Delay (D) is expressed as the sum of delays (*di*), divided by the total number of all measurement (N)

$$D = \sum_{i=1}^{N} \frac{di}{N}$$

B. Jitter (Variation of Delay)

In order for voice to be intelligible, voice packets must arrive at regular Intervals. Jitter describes the degree of fluctuation in packet access, which can be caused by too much traffic on the line [4]. Voice packets can tolerate only about 75 Milliseconds (0.075 sec) but is preferred be 40 Milliseconds (0.040 sec) of jitter delay [5]. Equation shows the calculation of jitter (j). Both average delay and jitter are measured in seconds. Obviously, if all (di) delay values are equal, then D = di and J = 0 (i.e., there is no jitter) [6].

$$\mathrm{J{=}f{=}}\frac{1}{\mathrm{N\text{-}1}}+\sum_{\mathrm{i=1}}^{\mathrm{N}}(\mathrm{di\text{-}D}) * (\mathrm{di} - \mathrm{D})$$

C. Packet loss

Packet loss is the term used to describe the packets that do not arrive at the intended destination that happened when a device (router, switch, and link) is overloaded and cannot accept any incoming data at a given moment [7]. Packets will be dropped during periods of network congestion. Voice traffic can tolerate less than a 3% loss of packets (1% is optimum) before callers feel at gaps in conversation [5]. Equation shows the calculation of packet loss ratio defined as a ratio of the number of lost packets to the total number of transmitted packets Where N equals the total number of packets transmitted during a specific time period, and NL equals the number of packets lost during the same time period [6].

$$\text{Loss packets ratio} = (NL / N) \times 100\%$$

III. CONGESTION MANAGEMENT

Congestion management features allow you to control congestion by determining the order in which packets are sent out an interface based on priorities assigned to those packets. When the network is designed to service widely varying types of traffic, there is a way to treat contention for network resources by queuing, and manages the available resources according to conditions outlined by the network administrator. Each router, as part of the resource allocation mechanisms, must implement some queuing (algorithm) discipline that governs how packets are buffered while wait to be transmitted. Congestion management entails the creation of queues, assignment of packets to those queues based on the classification of the packet, and scheduling of the packets in a queue for transmission. The congestion management QoS feature offers four types of queuing protocols, each of which allows you to specify creation of a different number of queues, affording greater or lesser degrees of differentiation of traffic, and to specify the order in which that traffic is sent. There are three common queuing disciplines that can be analyses, they are first-in-first-out (FIFO) queuing, priority queuing (PQ) and weighted-fair queuing (WFQ) [8]. The basic principle of FIFO queuing is that the first packet that arrives at a router is the first packet to be transmitted. An exception here happened if a packet arrives and the queue is full, then the router ignores that packet at any conditions. [8]. As shown in the Figure 1.

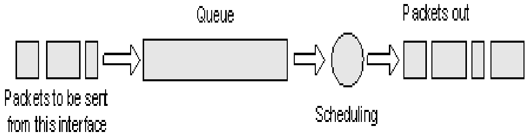


Fig 1: FIFO queue

The principle idea of PQ queuing depends on the priority of the packets, a highest priority are transmitted on the output port first and then the packets with lower priority and so on. When congestion occurs, packets with lower-priority queues will be dropped. The only problem with these packets is that has lower-priority in queue [9]. As shown in the figure2.

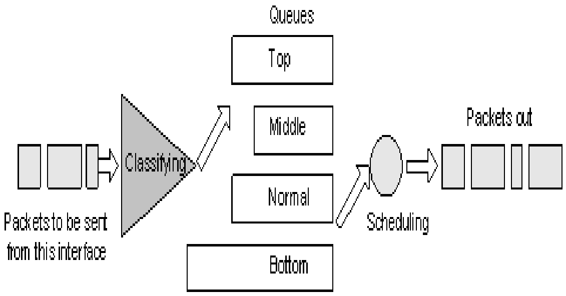


Fig 2: Priority queuing

The Weighted-fair queuing discipline provides QoS by provides fair (dedicated) bandwidth to all network traffic for control on jitter, latency and packet loss. The packets are classified and placed into queues according to information ToS field in IP header is use to identify weight (bandwidth). The Weighted-fair queuing discipline weights traffic therefore low-bandwidth traffic gets a high level of priority. A unique feature of this queuing discipline is the real-time interactive traffic will be moved to the front of queues and fairly the other bandwidth shares among other flows [9]. As shown in the Figure3.

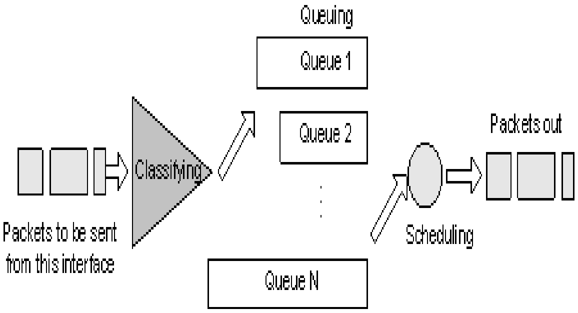


Fig 3: WFQ queuing

IV. CONFIGURING CBWFQ and LLQ

Class Based weighted Fair Queuing (CBWFQ) - CBWFQ is most like CQ, in that it can be used to reserve minimum bandwidth for each queue. It does differ from CQ in that you can configure the actual percentage of traffic, rather than a byte count. CBWFQ is like WFQ in that CBWFQ can actually use WFQ inside one particular queue, but it differs from WFQ in that it does not keep up with flows for all the traffic. CBWFQ classifies packets using the exact same set of fields that "MQC, QPM, and AutoQoS." CBWFQ's use of MQC makes learning the configuration for CBWFQ easy, assuming you remember how to configure CB marking from the preceding chapter. And unlike WFQ, which use flow-based classifiers, CBWFQ does not classify based on the flow, but on anything you can match with the MQC commands.

CBWFQ supports 64 queues, with a maximum and default queue length varying depending on the model of router and the amount of memory installed. All 64 queues can be configured, but one class queue, called *class-default*, is automatically configured. If the explicitly configured classification does not match a packet, IOS places the packet into the class-default class. You are allowed to change the configuration details regarding this default class, but this one class always exists. So far, the other queuing tools in this chapter supported only FIFO logic inside a single queue. Currently, CBWFQ can use either FIFO or WFQ inside the class-default queue. With Flow-Based WFQ in the class-default queue, when CBWFQ decides to take one or more packets from the queue, it takes the packet with the best sequence number (SN) — just like WFQ normally does. CBWFQ provides a great advantage by allowing WFQ to be used in the class-default queue. You may recall that WFQ is actually a very good default choice for queuing, because it treats low-volume flows well, and many low-volume flows are also interactive flows. WFQ also treats packets with high precedence well. So, with CBWFQ, for the traffic you know about, you classify it, and reserve the right amount of bandwidth for the class. For the traffic you cannot characterize, you let it default into the class-default queue, where you can dynamically apply some fairness to the default traffic by using WFQ. The capability to reserve bandwidth for some packets, and fairly assign the rest of bandwidth with WFQ, makes CBWFQ a very powerful queuing tool.

Table1: Configuration CBWFQ

CBWFQ Feature	Description
Classification	Classifies based on anything that MQC Commands can Match, just like CB marking. Includes all extended IP ACL fields, NBAR ,incoming interface, CoS ,Precedence ,DSCP ,source/destination MAC,MPLS Experimental ,QoS group, and RTP port numbers.
Drop policy	Tail drop or WRED, Configurable per queue
Number of queues	64
Maximum queue length	Varies per router model and memory
Scheduling inside a single queue	FIFO on 63 queue, FIFO or WFQ on class-default queue
Scheduling among all queues	Algorithm is not published .The result of the scheduler provides a percentage guaranteed bandwidth to each queue

Low Latency Queuing (LLQ) - LLQ is not really a separate queuing tool, but rather a simple option of CBWFQ applied to one or more classes. CBWFQ treats these classes as strict-priority queues. In other words, CBWFQ always services packets in these classes if a packet is waiting, just as PQ does for the High queue. LLQ uses the priority command instead of the bandwidth command to request bandwidth. The priority command guarantees that the requested bandwidth is available whether the interface is busy or not. Because this bandwidth is always available, the class map that uses the priority command is guaranteed low latency through the interface (thus the name, LLQ). This is also called a strict priority queue. LLQ scheduler always checks the low-latency queue first, and takes a packet from that queue. If there are no packets in the low-latency queue, the normal, scheduler logic applies to the other non-low-latency queue queues, giving them their guaranteed bandwidth. For delay-sensitive traffic, the addition of a low-latency queue overcomes the one big negative of CBWFQ. In fact, with all the other queuing tools covered in this chapter so far, only PQ gave voice traffic the best quality. Of course, PQ had the negative side effect of almost destroying the performance of the lower-priority applications when the link was congested. With LLQ, you get the best of both worlds — low latency for the traffic in one queue, and guaranteed bandwidth for the traffic in other queues. If you follow these lines, you can see a path through the logic for LLQ in which only the low-latency queue gets any service. How can LLQ guarantee the other queues their respective bandwidths, with logic that never lets those queues get serviced? LLQ actually *polices* the PQ based on the configured bandwidth. By doing so, the packets in the queue that are forwarded still have very low latency, but LLQ also prevents the low-latency traffic from consuming more than its configured amount of bandwidth. By discarding excess traffic, LLQ can still provide bandwidth guarantees to the non-priority queues. The policing function of LLQ takes care of protecting the other queues from the low-latency queue, but it does discard packets to accomplish that goal. Take a moment to reflect on the types of traffic that need to be classified into the low-latency queue. VoIP traffic, and in most cases, video traffic, need the low-latency, low-jitter performance of the low-latency queue. However, these are the same types of traffic that are most sensitive to dropped packets. So, although putting voice and interactive video into the low-latency queue may be good for queuing, discarding packets that exceed the configured rate for the queue would be harmful to those types of traffic.

Table 2: Configuration LLQ

Command	Purpose
Router(config-pmap-c)# **priority** *bandwidth*	Reserves a strict priority queue for this class of traffic.

Router(config-if)# **max-reserved-bandwidth** *percent*	Changes the maximum configurable bandwidth for CBWFQ, LLQ, and IP RTP Priority. The default is 75 percent.

V. COMPARING TWO DIFFERENT QUEUING MECHANISM CBWFQ and LLQ

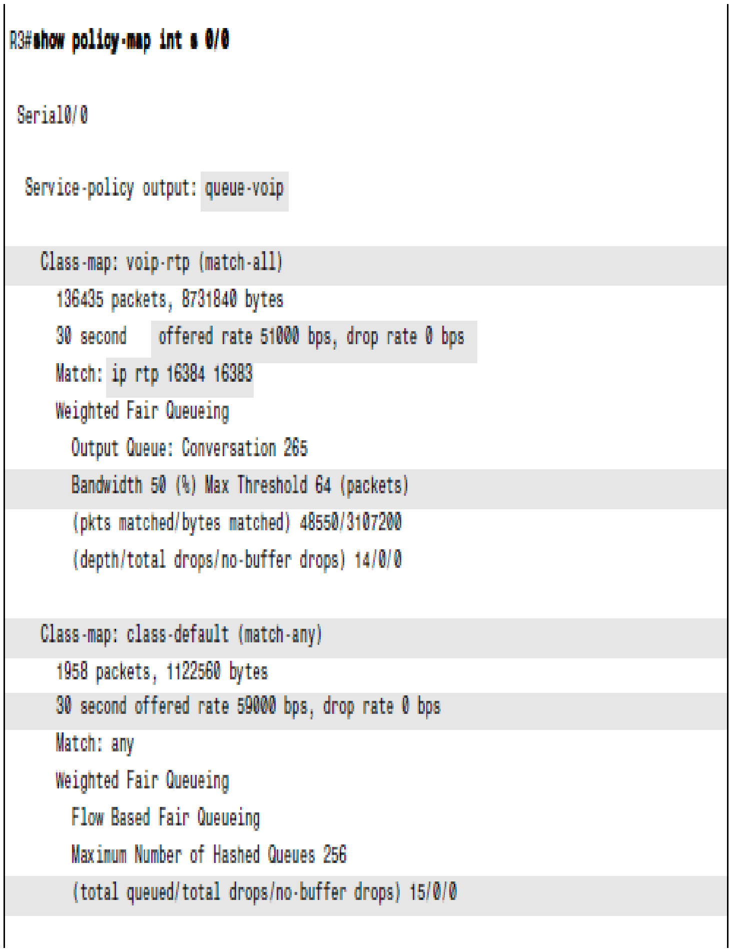
CBWFQ (being a subsequent of WFQ) and LLQ each have an important role to play with QoS implementations in Cisco routers. WFQ works well as a default queuing mechanism when there are no low-latency requirements due to its very simple configuration and fair treatment of typically more important low volume flows. When an engineer takes the time to figure out what types of traffic need particular levels of service, CBWFQ provides bandwidth reservation with each class. When those types of traffic include classes with low latency and low jitter requirements, LLQ allows bandwidth reservation, priority service, with protection for the lower priority queues.

The below table represents the more important points about these queuing tools, with comments about their support of each point:

Table 3: Comparing CBWFQ and LLQ

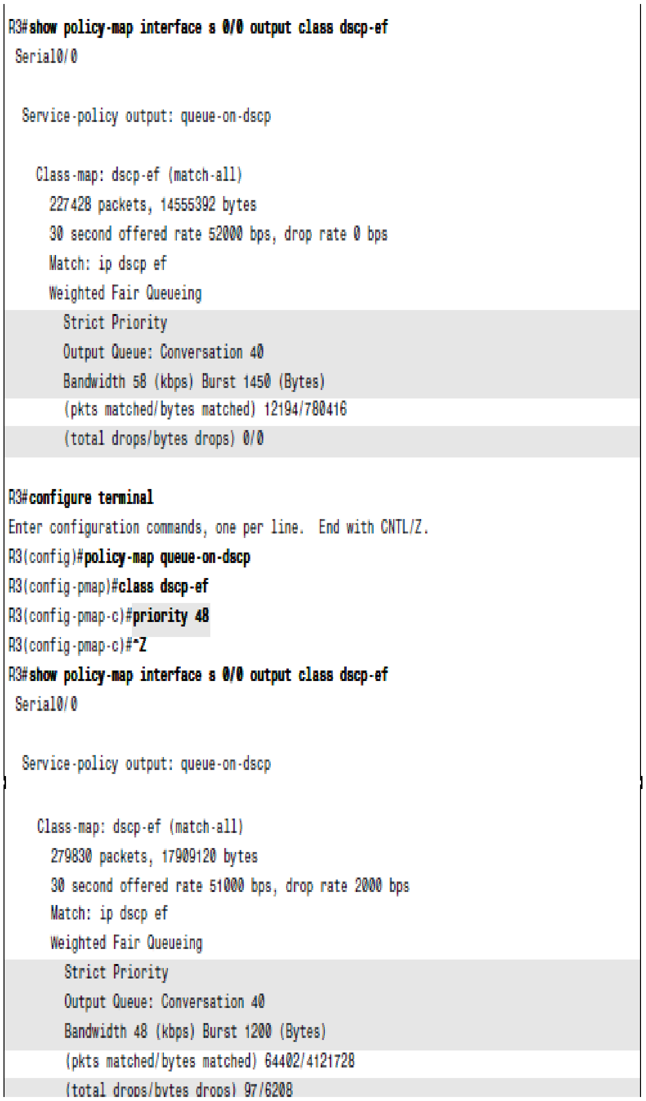
Concept	**CBWFQ**	**LLQ**
Requires complex Classification Configuration	Yes	Yes
Uses MQC	Yes	Yes
Prefer low Volume,high Precedence Flows	Not flow based	Not flow based
Experiences problems with large numbers of flows	No	No
Can reserve bandwidth per Queue	Yes	Yes
Provide low delay,low jitter queuing	No	Yes

VI. RESULT



```
R3#show policy-map int s 0/0

 Serial0/0

  Service-policy output: queue-voip

    Class-map: voip-rtp (match-all)
      136435 packets, 8731840 bytes
      30 second   offered rate 51000 bps, drop rate 0 bps
      Match: ip rtp 16384 16383
      Weighted Fair Queueing
        Output Queue: Conversation 265
        Bandwidth 50 (%) Max Threshold 64 (packets)
        (pkts matched/bytes matched) 48550/3107200
        (depth/total drops/no-buffer drops) 14/0/0

    Class-map: class-default (match-any)
      1958 packets, 1122560 bytes
      30 second offered rate 59000 bps, drop rate 0 bps
      Match: any
      Weighted Fair Queueing
        Flow Based Fair Queueing
        Maximum Number of Hashed Queues 256
        (total queued/total drops/no-buffer drops) 15/0/0
```

Fig 4: Packet transferred using CBWFQ



```
R3#show policy-map interface s 0/0 output class dscp-ef
 Serial0/0

  Service-policy output: queue-on-dscp

    Class-map: dscp-ef (match-all)
      227428 packets, 14555392 bytes
      30 second offered rate 52000 bps, drop rate 0 bps
      Match: ip dscp ef
      Weighted Fair Queueing
        Strict Priority
        Output Queue: Conversation 40
        Bandwidth 58 (kbps) Burst 1450 (Bytes)
        (pkts matched/bytes matched) 12194/780416
        (total drops/bytes drops) 0/0

R3#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
R3(config)#policy-map queue-on-dscp
R3(config-pmap)#class dscp-ef
R3(config-pmap-c)#priority 48
R3(config-pmap-c)#^Z
R3#show policy-map interface s 0/0 output class dscp-ef
 Serial0/0

  Service-policy output: queue-on-dscp

    Class-map: dscp-ef (match-all)
      279830 packets, 17909120 bytes
      30 second offered rate 51000 bps, drop rate 2000 bps
      Match: ip dscp ef
      Weighted Fair Queueing
        Strict Priority
        Output Queue: Conversation 40
        Bandwidth 48 (kbps) Burst 1200 (Bytes)
        (pkts matched/bytes matched) 64402/4121728
        (total drops/bytes drops) 97/6208
```

Fig 5: Packet transferred using LLQ

VII. CONCLUSION

The CBWFQ is a mechanism that is used to guarantee bandwidth to classes. It guarantees bandwidth according to the weights assigned to traffic classes. A queue is reserved for each class and traffic belonging to a class is directed to that class queue. The 75% rule of the bandwidth here will be assigned to major applications such as voice , video and data. However, LLQ feature brings strict priority queuing to CBWFQ approach only, say by adding a priority queue to CBWFQ for real time traffic. So, CBWFQ is a modern congestion management technique and it is not advised for Voice and Video traffic, since both of them need strict priorities. With CBWFQ you can define traffic classes and assign guaranteed amount of minimum bandwidth, these classes can get more bandwidth if its available, but they always get the minimum bandwidth assigned to them. But, LLQ is a modern congestion management technique, which is extending CBWFQ to support Voice and Video or mission critical traffic. LLQ is adding Priority Queuing to the CBWFQ. The Priority Queue is used only for Voice / Video or mission critical traffic, without having the Queue Starvation for other Queues. The starvation is avoided using the policing, the traffic in Priority Queue is policed and the Queue can get to whatever the bandwidth it was assigned. It can however go over the assigned bandwidth if there is no congestion. The strict policing applies only in times of congestion and that too to the traffic in the Priority Queue so it does not starve other queues. The mission critical traffic gets the bandwidth it is assigned and other queues get whatever minimum bandwidth they were assigned, plus if there is no congestion all the queues can go over their assigned bandwidth if needed. The Voice traffic is handled by the PQ Scheduler and the remaining traffic is handled by the WFQ handler. The policed bandwidth for traffic in PQ is only during the time of congestion, if there is no congestion then the traffic in PQ (typically voice) can go over the bandwidth assigned to it. The key disadvantages with CBWFQ are that delay in real time traffic might occur and no mechanism exists to provide a strict-priority queue for real-time traffic, such as VoIP, to alleviate latency and LLQ addresses it.

REFERENCES

[1] "End-to-end quality of service support over heterogeneous networks". Project description. European Community Research and Development Information Service. Retrieved October 12, 2011.

[2] Haniyeh Kazemitabar, Sameha Ahmed, Kashif Nisar, Abas B Said, Halabi B Hasbullah , “A Survey on Voice over IP over Wireless LAN” , Paper in World Academy of Science, Department of Computer and Information Sciences, University Technology, Malaysia , 2010 .

[3] Greg S. Tucker, “Voice over Internet Protocol (VoIP) and Security”, Paper in SANS Institute, October 2004.

[4] Gouda I. Salama, M. Elemam Shehab, A. A. Hafez, M. Zaki, “Performance Analysis of Transmitting Voice over Communication Links Implementing IPSec ” , Paper in 13th International Conference on Aerospace Sciences and Aviation Technology (ASAT), Military Technical College, Cairo, Egypt , May 2009.

[5] Qinxia (Alice) He "Analyzing the Characteristics of VoIP Traffic”, Thesis Msc, Department of Computer Science, University of Saskatchewan, Canada, July 2007.

[6] Furat Nidhal Tawfeeq,“Network Congestion and Quality of Service Analysis Using OPNET”, Thesis, Department of Information Engineering, Al-Nahrain University, March 2009.

[7] Jishu Das Gupta, “Performance Issues for VoIP in Access Networks”, Thesis Msc, Department of Mathematics and Computing Faculty of Sciences, University of Southern Queensland, 2005.

[8] Fulvio Ricciardi. "QoS and Traffic Shaping in Transparent Bridge mode". Router/Bridge Linux Firewall website. ZeroShell Net Services. Retrieved October 15, 2011

[9] http://www.cisco.com/en/US/prod/collateral/iosswrel/ps6537/ps6558/sol_ov_c22-708224.html