ISSN 2229-5518

An Overview of Quality of Service Architectures and Congestion Management Techniques in Ip Networks

Ogbonna, Ugochukwu

Abstract — Data networks worldwide are faced with the dilemma of huge amounts of data transfers. With such large amounts being exchanged over such networks, it becomes necessary to devise strategies for management of such voluminous transmissions. Quality of service defines sets of rules which monitor and prioritize data thereby ensuring that packet/data handling and arrival sequence is done based on priority. This paper serves to present an investigative review into the various current architectures used to deploy quality of service approaches in IP networks, as well as the various congestion management techniques used to realize management of data congestion in such networks.

Index Terms —Congestion Management, DiffServ, IntServ, IP Networks, MPLS, Quality of Service

1 INTRODUCTION

The emergence of converged and heterogeneous networks, unified communications, and the ongoing internet of things have resulted in more diverse data being sent over public and private IP networks. Moreover, the standard for data delivery was the best effort delivery system, where data packets are transmitted on a "First-in First-out" basis, with no guarantee of successful delivery. However, this technique usually proves to be inadequate and presents competency issues, especially where real-time traffic are involved. This is because different applications demand different network services, which may in turn, make them tolerate some level or types of network impairment, whereas other applications may not (e.g. some application may tolerate delays in data transmission while others may not). Hence, there is a need for a mechanism that can manage data transmission as per application or design requirements.

Quality of Service (QoS) defines a set of approaches which is aimed at providing guarantees on the ability of a network to deliver predictable results of delay , bandwidth, packet loss and jitter [8]. Generally speaking, it is the ability of a network element (e.g. an application, a host or a router) to provide some level of assurance for consistent network data delivery, while managing traffic congestion and measuring service availability and transmission. Given that networks today are ubiquitous and pervasive, and susceptible to undifferentiated and unpredictable service responses [5], a mechanism such as QoS is required to specify guaranteed network data performance level, and ensure that data traverse the network satisfactorily. Providing flexible end-to-end QoS implies the ability to manage delay and available bandwidth to individual data flow [2]. In the following sections of this paper, we will examine the existing QoS Architectures, after which the various congestion management techniques deployed in QoS will then be considered. The last section of the chapter then concludes the paper, pointing out areas which may require additional study.

U. Ogbonna obtained his BEng degree in Electrical/Electronic Engineering from Federal University of Technology, Owerri, Nigeria and MSc in Telecommunications Engineering from University of Sunderland, UK. His research interests include Quality of Service/Experience in homogenous networks, protocol analysis and energy-efficiency management in enterprise LANs and wireless networks, and design and management of data/multimedia inter-activity in unified communication networks. Currently, he is with the Department of Electrical/Electronic Engineering Michael Okpara University of Agriculture, Umudike, Nigeria. Email:ugogbos99@yahoo.co.uk

2 QoS ARCHITECTURES

As previously stated, the global nature of applying data networks could result in vulnerable and unacceptable service responses. Given that networks traditionally deliver traffic using best-effort delivery, QoS strategies need to be devised to ensure seamless co-operation and interactivity of data and multimedia traffic. As a result, the Internet Engineering Task Force has standardized three QoS model frameworks for designing QoS strategies; the Integrated Services model (*IntServ*) and Differential Services model (*DiffServ*), and Multiprotocol Label Switching (MPLS) [2], [4].

2.1 The Integrated Services (IntServ) Model

The IntServ approach attempts to deliver end-to-end QoS guarantees by establishing an end-to-end connection for each flow and maintaining the state of all connections. It was introduced in IP networks to provide guaranteed and controlled services in addition to the existing best effort service [14]. It is based explicitly on signalling and management/reservation of network resources for applications requesting for it. It was the first model proposed by the IETF as a solution to two fundamental challenges:

- Most real-time applications do not function optimally across the Internet due to existence variable queuing delays and losses due to congestion, and
- Operators usually lack the capacity to control the bandwidth distribution on specific links when different traffic classes exist [4].

In the IntServ framework, end-to-end QoS guarantee is

ISSN 2229-5518

achieved using per flow based, hop-by-hop signalling. This is possible through the use of Resource Reservation Protocol (RSVP). RSVP is a transport layer protocol occupying the fourth layer of the OSI model. It is designed to work with existing and future routing protocols, and consults existing routing tables to obtain accessible routes in data networks. RSVP reserves bandwidth at routers along the data flow path. When data arrive at the ingress edge router (IER) with a QoS necessity, the IER initiates the required path establishment process by sending a PATH message to the destined egress edge router (EER). The EER then returns an RESV message back to the IER and tries to reserve the required bandwidth so to meet the requested QoS along the traffic path to the source IER. Routers along this path adjust their traffic control mechanisms such that each admitted flow is guaranteed to receive the reserved bandwidth, thus accomplishing the requested QoS.

IntServ provides guaranteed QoS, delivering end-to-end resource and pre-request policy admission control. This ensures maintenance of necessary QoS requirements to the applications as long as the data packets are within the specified traffic parameters [8]. However, IntServ nodes have to maintain, save and update all flow-states, including router implementation of RSVP, packet classification, scheduling, policing and queuing, especially when a network is large [2], [8]. Moreover, per-flow QoS is difficult to realize in an end-to-end network without significant increases in complexity and cost. This need for flow tracking and maintenance presents IntServ with scalability challenges, and also leads to increased, overhead with increased flow as each flow requires continuous signalling [2].

2.2 The Differentiated Services (DiffServ) Model

For The DiffServ QoS framework employs categorization and prioritization network traffic (flow) into aggregates. It was proposed to deliver scalable and manageable, packet differentiation capability and services for IP networks [14]. In this model, individual data flows are aggregated into traffic classes at the various edge routers and core routers proceed to forward received data packets to the next hop according to the Per-Hop Behaviour (PHB) associated with the traffic class. The IETF has introduced 3 main categories of PHB classes namely;

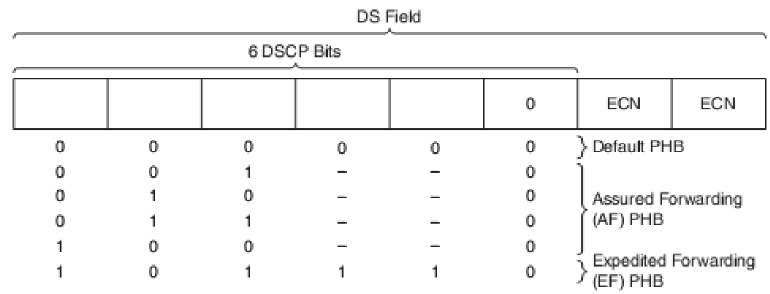
Default Forwading: This PHB is used to map the already existing best-effort forwarding behaviour used in the Internet, to the DiffServ model. It ensures backward compatibility with the non-DiffServ supporting nodes. This is achieved by setting the DSCP value of the DiffServ field (DS field) in the IP headers of data packets to “000000”.

Assured Forwarding: This PHB class is defined for traffic not deemed extremely critical, but cannot be simply discarded. It is further divided into 12 sub-classes, with each class composed of 3 PHB’s arranged to represent different levels of drop precedence. Packets with higher category and DP value are given greater priority. According to [4], there are 3 traits that affect the level of forwarding assurance given to an IP packet. These aspects are:

- The “quantity” of resources allocated to the AF class of the packet;
- The current load of the class; and
- The drop precedence of the packet in the case that the class is congested.

Table 1 – Table showing the various DiffServ AF Classes

Drop Precedence (DP)	Low DP	Medium DP	High DP
Class #1	AF11 (001010)	AF12 (001100)	AF13 (001110)
Class #2	AF21 (010010)	AF22 (010100)	AF23 (010110)
Class #3	AF31 (011010)	AF32 (011100)	AF33 (011110)
Class #4	AF41 (100010)	AF42 (100100)	AF43 (100110)



Expedited Forwarding: this PHB service model offers low delays, low losses, low jitters, assured bandwidth, and end-to-end services through DiffServ domains. It corresponds to a decimal value of 46, which involves setting the DSCP value of the DS field to “101110”.

Figure 1: IP Header showing DS field and DSCP bit values

DiffServ promotes scalability by aggregating individual flows in to a smaller number of traffic classes and providing differing treatments for different traffic classes. Here, traffic classification according to designed PHB’s can be done at either the second or third layers of the OSI model. However, layer 3 classification finds greater application as layer 2 traffic markings cannot be preserved end-to-end [1]. In DiffServ, flows are grouped into a small number of classes at network boundaries, and the routers within the network merely implement a set of scheduling processes based on these classes. This per class tactic streamlines the router’s functionalities and reduces the flow states that routers have to retain. Layer 3 classification involves border routers classifying incoming packets according to the class-based specified criteria, and establishing IP precedence (IPP) to different packet classes [8]. The packets are then assigned to specific traffic classes using the Differentiated Service Code Point (DSCP). DSCP is a 6-bit IP-header marking, which classifies/aggregates packets, defining priority levels and service types on a PHB at each network node along the path. Traffic with similar DSCP belonging to same PHB are grouped together as Behaviour Aggregates (BAs), and the network then executes the scheduled strategies designed for each BA.

Overall network scalability is improved by providing services on a per-class basis instead of a per-flow basis, which also removes the per-flow QoS signalling overhead, and the core routers not retaining flow states as opposed to IntServ framework, thus making DiffServ more scalable. However DiffServ does not provide end-to-end QoS guarantees to individual flows, in addition to its inability to exercise admission control at the network edge. When overload conditions occur in a given service class, all flows in that class suffer potentially harsh service degradation regardless of the QoS requirements of the class. However, its incapacity to provide guaranteed end-to-end to individual flows like IntServ could be overcome through a combined application of Committed Access Rate (CAR) technology, which involves packet classification according to IP Precedence, and Queuing technology for queuing control and congestion management [8].

2.3 The Mulit-Protocol Label Switching (MPLS) Model

Traditionally, routers' packet-forwarding decisions are done through routing information look up based on the destination IP address in each packet. Routers along the data path analyse and forward IP packets along a routed path from one router to the next they reach their destination. The choice of the next hop for each packet is as a result of running a biggest prefix match search routing algorithm involving IP header analysis, and then an additional forwarding algorithm in the network layer. However, because routers can occasionally prove to become QoS bottlenecks, this technique of data delivery can be sometimes complex and insufficient to support some network demands, even when IP Precedence and DiffServ schemes are employed [7].

MPLS was initially introduced as a way of creating Virtual Private Networks (VPNs) over a common IP backbone, but was then later standardized as a mechanism for improving the forwarding speed of routers [4]. It circumvents the need for header-lookup thereby eliminating complexities arising from that necessity, operating at a "sub layer" between layers 2 and 3 of the OSI model and bringing transmission speeds in layer 2 up to that of layer 3 [7]. The idea of MPLS was to assign short, fixed-length identifiers called routing labels, to packets to simplify and increase forwarding process while implementing hop-by-hop routing. Labels are used as a type of index in a table specifying next hop directions, with all the path forwarding decisions based on the associated label. They are assigned based on parameters like priority and destination, and attached to packets at border routers. With this switching method, routers forward based on label contents rather than destination IP address lookup, and this feature makes MPLS perform as some sort of QoS structure.

MPLS brings a number of other benefits to IP-based networks including RSVP-like guaranteed QoS; nevertheless, though few MPLS networks are actually functioning today because the specifications are still slightly in flux. However, [4] state that MPLS does not necessary introduce a new QoS framework. This is backed by [9] who advocated the importance of implementing additional QoS support to preserve desired service quality. A standard presented by IETF in an effort to realize such strategy is the DiffServ-Aware-MPLS with traffic engineering (MPLS-TE) which combines IntServ and DiffServ features to differentiate traffic and create service models realising the possibility of a more manageable QoS delivery [4]. TE enables efficient network traffic management while considering link availability. IntServ is applied at network edges where scalability is not considered foremost priority, while DiffServ is applied in the network core where large flows exist and scalability is a primary goal. Admission control of individual flows and service mapping from IntServ service type to DiffServ traffic class are the responsibility of edge routers. Network core routers recognize traffic classes and not individual flows, and also may or may not be RSVP aware. Although MPLS-TE could achieve better QoS enhancement and optimization, it could also add unnecessary complexities to existing QoS composition [4].

3 Congestion Management

QoS is a network mechanism that manages data transfer deluges in networks. With the extensive implementation of heterogeneous networks and unified communications, there always is the possibility of experiencing data speed-mismatch and aggregations at network edges, which inadvertently results in traffic congestion. Congestion can cause jitters and increase overall network latency in data networks.

QoS facilitates control of data transmission quality in networks, while enhancing data traffic flows travelling through many different network technologies [6]. It introducing queuing mechanisms which creates a means of managing traffic congestion and controlling packet transmission and drop. According to [3] and [6], QoS queuing techniques can be commonly categorized into the following;

- First-In-First-Out (FIFO)
- Priority Queuing (PQ)
- Weighted Fair Queuing (WFQ)
- Class-Based Weighted Fair Queuing (CBWFQ), and
- Low Latency Queuing (LLQ)

3.1 First In First Out (FIFO)

FIFO defines the most basic behaviour of a network to handle traffic queues on a first come first served basis. In other words, what comes in first is handled first, with the next queue waiting until the first is finished. This is the default queuing technique in most interfaces, and is usually implemented when no particular queuing technique is specified. Under this technique, all packets are treated equally by placing them into a single queue, then servicing them in the same order they were placed in the queue.

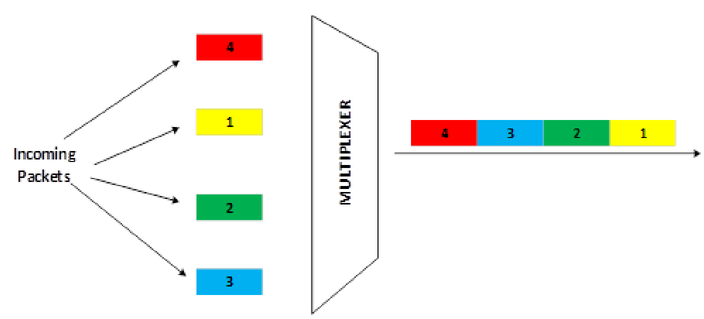


Figure 2: Graphical illustration for FIFO technique

The packets are given equal priority and attended to in order of arrival. However, due to absence of drop and scheduling

ISSN 2229-5518

mechanisms, it can be susceptible to jitter and bandwidth consumption by high-capacity applications [12]. Router vendors, in a bid to manage queuing, occasionally implement two queues on an output port when queue scheduling strategies are absent – a high-priority queue dedicated to managing traffic network control, and FIFO technique that handles all other types of traffic.

3.2 Priority Queuing (PQ)

PQ is a simple variation of FIFO queuing. This technique involves marking packets with priority tags so that routers can implement multiple FIFO on each marked packet. Four priority-queue hierarchies namely; high, medium, normal and low are defined, and each packet is assigned to a particular hierarchy depending on the priority value assigned to it [12]. Each hierarchy is then treated as FIFO from the highest hierarchy to the lowest, with each hierarchy serviced when the packets belonging to the hierarchy with higher priority are empty. Packets are then forwarded until either packets within the current hierarchy are exhausted or packets arrive at a higher hierarchy.

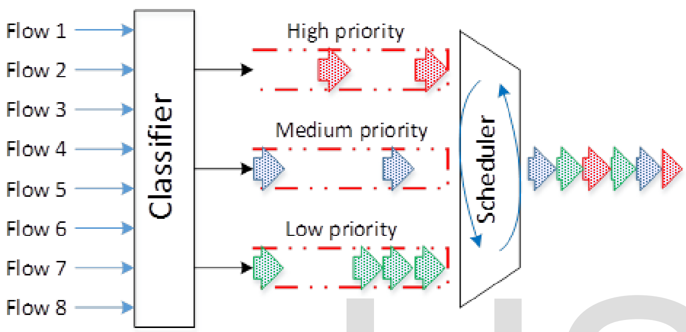


Figure 3: Graphical illustration for PQ technique

PQ is the basis of queue scheduling techniques which provides a simple support for DiffServ classes [6]. This technique however, could result in the "starvation" of service or non-transmission of lower-hierarchy packets if the higher-hierarchy receives steady packet streams.

3.3 Weighted Fair Queuing (WFQ)

WFQ is a flow-based queuing algorithm introduced to overcome delay and jitter issues with FIFO, and starvation issues with PQ [1]. The idea behind this method is to maintain distinct queues for each traffic flow handles processed by the router. It allocates bandwidth fairly to network traffic, using various classification-bases as weights to determine the amount of bandwidth for various traffic-flows. The flows are assigned to a FIFO queue using a WFQ scheduler, which assigns finish-times to packets. Packets with smaller finish-times are then transmitted first. After each packet is classified and slotted into its necessary queue, the scheduler then estimates and assigns a finish time for the packet. As each queue is examined by the WFQ scheduler, the packet with the smallest (earliest) finish time is chosen as the next packet for transmission through the output port.

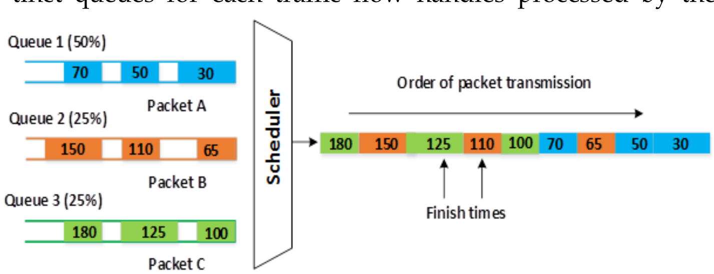


Figure 4: Graphical illustration for WFQ technique

WFQ is simple to configure, and guarantees throughput with the absence of flow-starvation as can be experienced in PQ. However, there is a possibility of assigning multiple traffic-flows to one queue, and no fixed traffic-flow guarantee.

3.4 Class Based Weighted Fair Queuing Queuing (CBWFQ)

CBWFQ is a topical scheduling mechanism intended for handling congestions while providing greater flexibility. Here, distinct traffic classes are defined by the network administrator using selected criteria, and specific minimal bandwidths assigned to each defined class. Packets matching each defined class form traffic for the class, and CBWFQ then guarantees bandwidth to the traffic classes, extending WFQ from individual traffic-flows to user-defined classes [3]. Depending on the criteria defined by the network administrator, each traffic class is put in a reserved queue and allocated a user-defined guaranteed bandwidth. However, unlike priority queuing, if more bandwidth is available, it makes use of it to push the queue out.

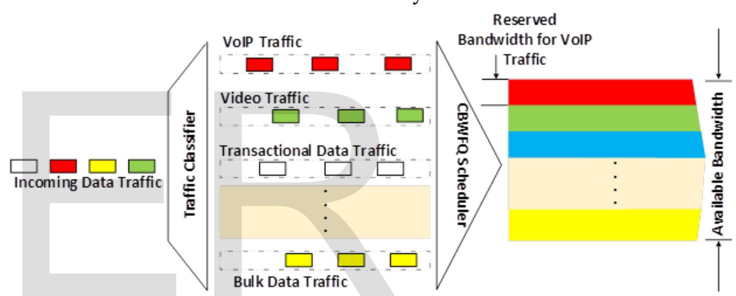


Figure 5: Simple illustration for CBWFQ technique

CBWFQ, through permitting the establishment of user-defined traffic classes, offers efficient bandwidth management, maximizes priority traffic transport and queue management thereby eliminating the challenges encountered with WFQ. It finds applications in situations where there is need to provide suitable bandwidth for specific applications. However, its queuing method is not suitable for real-time applications since they have no priority-queuing mechanisms [12].

3.5 Low Latency Queuing (LLQ)

LLQ combines the attributes of PQ and WFQ/CBWFQ, introducing high-priority queuing together the traffic classification property of WFQ/CBWFQ. Here, traffic classified as strict-

ISSN 2229-5518

priority traffic class is allocated guaranteed bandwidth. It allows high-priority traffic receive priority treatment while residual bandwidth is shared across other traffic types using WFQ/CBWFQ. LLQ attempts to guarantee minimal delay of real-time traffic, and is currently recommended for networks running VoIP and multimedia applications. There is usually one Priority Queue and a couple of Weighted Fair Queues, and real-time traffic is queued to the priority queue while all other traffic is allocated to the WFQ/CBWFQ priority queues. In this way, it minimizes delays and jitters in transmission of real-time traffic. The priority queue is processed ahead of other queues, thereby allowing real-time data to be sent across the network to the receiving end as quickly as possible.

The LLQ scheduler constantly examines the network for high priority packets in priority queues, and ensures early processing and departure of such data packets. However, if LLQ packets are absent, the network then applies normal scheduling logic to the remaining queues. If two flows area assigned similar priority levels, they will be processed according to WFQ/CBWFQ policy. Furthermore, if the priority traffic exceeds its allocated bandwidth, it is policed and dropped. In other words, as long as the priority packets class do not exceed the set bandwidth limit, it gets LLQ-processed with marginal delay. This prevents starvation of lower-priority traffic.

4 CONCLSION

This paper was aimed at reviewing currently existing QoS strategies and models which have been standardized, and are being deployed in networks to manage data transmission. With data management and network administration a core priority for many enterprise networks, additional models and techniques are currently being researched. There are also attempts to combine to combine two models in an attempt to take advantage of the benefits of each model. This has given rise to architectures such as MPLS-TE and DiffServ-MPLS mapping. In this paper has however, we have limited its review to the existing standards proposed by the IETF for designing QoS architectures, and devising approaches to control data congestion.

It is also be worthy to mention that achieving complete QoS management in networks may not be effectively realized in some situations without the consideration of adding appropriate congestion avoidance techniques such as Random Early Detection (RED) and Weighted RED (Peculea et. al). By default, networks apply tail drop approach, dropping packets arriving latest when queues are full. Congestion management techniques defines circumstances for which packets can be dropped, and the order of priority for dropping them, thereby ensuring that more precedence is assigned to priority packets in the event of a filled up queue.

REFERENCES

[1] Cisco (2006). Implementing Cisco Quality of Service, Vol.1-2, Version 2.2. San Jose, Carlifonia: Cisco Press.

[2] Fgee, E. B., Kennedy, J. D., Philips, W. J., Robertson, W., & Sivakumar, S. (2005). Comparison of QoS Performance between IPv6 QoS Manangement Model and IntServ and DiffServ QoS Models. Proceedings of the 3rd Annual Communications Networks and Services Research Conference, 2005, 287-292.

[3] Fischer, M. J., Masi, D. M. B. & Shortle, J. F. (2008). Approximating Low Latency Queueing Buffer Latency. In: Telecommunications, 2008. AICT '08. Fourth Advanced International Conference on, 8-13 June 2008 (2008). 188-194.

[4] Flavius, P. and Ferdi, P. (2011). Internet Service Delivery Models: Evolution and Current Issues. International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), 2011 146-153.

[5] Hoang, D. B. and Phan, H. T. (2007). A scalable end-to-end QoS architecture. International Symposium on Communications and Information Technologies, 2007. ISCIT '07., 677-682.

[6] Islam, M. Z., Islam, M. S., Haque, A. K. M. F. and Ahmed, M. (2012). A comparative analysis of different real time applications over various queuing techniques. Informatics, Electronics & Vision (ICIEV), 2012 International Conference on, 1118-1123.

[7] Kelly, E. B. (2002). Quality of Service in Internet Protocol (IP) Networks. International Communications Industries Association, Infocomm, 1-5.

[8] Lipu, M. S. H., Haque, M. N. U., Karim, T. F., Rahman, M. L. and Morshed, M. H. (2010). Quality of service model selection criteria and traffic controlling algorithm of a native IP Network for telecommunication services. International Conference on Educational and Information Technology (ICEIT), 2010, 3, V3-144-V3-148.

[9] Molnar, K. and Vleck, M. (2010). Evaluation of Quality-of-Service Support in MultiProtocol Label Switching. Fifth International Conference on Systems and Networks Communications (ICSNC), 2010, 262-265.

[10] Odinma, A. and Oborkhale, L. (2006). Quality of Service Mechanisms and Challenges for IP Networks. The Pacific Journal of Science and Technology, vol.7 No.1, pp. 10-16.

[11] Raghupathikumar, D. and Bommanna Raja, K. (2014). A Combined Low Latency and Weighted Fair Queuing Based Scheduling of an Input-Queued Switch. Journal of Computer Science, Vol.10 (8), pp. 1447-1457.

[12] Ranjbar, A. (2007). *CCNP ONT official exam certification guide*, Cisco press.

[13] Yasin, W. and Ibrahim, H. (2010). Improving Triple Play Services Using Multi Protocol Label Switching Technology. Journal of Computer Science, vol 6 (3), pp. 269-278.

[14] Zhang, Q., Zhu, W., and Zhang, Y. (2004). End-to-End QoS for Video Delivery Over Wireless Internet. Proceedings of the IEEE, vol. 93 No. 1, pp. 123-131.