Channel Quality Based Adaptation of TCP with Loss Discrimination

Yaling Yang, Honghai Zhang, Robin Kravets
University of Illinois-Urbana Champaign

Abstract—**TCP responds to all losses by invoking congestion control and avoidance algorithms, resulting in degraded end-to-end performance in lossy environments. In recent years, two approaches have been taken to improve the performance of TCP in such networks. One is local retransmission and the other is end-to-end control. These schemes either impose a heavy computational burden on base station, or cannot achieve both efficient bandwidth usage and fairness between flows because they cannot distinguish the flows[1] on low quality wireless links from flows on high quality links.**

In this paper, we present a new wireless TCP protocol -- End-to-end Link State Aware TCP (TCP-ELSA) -- that adjusts the sending rate of a TCP flow according to the wireless link quality, but still provides good congestion control when congestion-related losses occur. The aim of TCP-ELSA is to adapt the sending rate of TCP to the link quality so that the efficiency of bandwidth usage is increased. We present simulation results that demonstrate the effectiveness of TCP-ELSA in various scenarios.

I. INTRODUCTION

Improvements in the reliability of wired communications have driven TCP to treat all losses as caused by congestion. The introduction of lossy wireless links breaks this assumption and causes degradation in the performance of TCP. Approaches to solving this problem are based on the observation that losses due to transmission errors should be retransmitted immediately and not be used as an indication of congestion. We refine this observation by noticing that retransmissions on a wireless link that is in a poor state (i.e. low SNR) may not be successful, but will consume resources on the wireless channel that might be better used by other flows. To address this problem, we present TCP-ELSA (TCP with End-to-end Link State Adaptation), which is a modification to TCP that bases retransmission decisions on information about the state of the wireless link when the loss occurred. TCP-ELSA combines loss discrimination techniques with end-to-end monitoring of loss rate.

In Section 2, we briefly describe the current solutions for improving TCP performance on lossy links and present the motivation of TCP-ELSA. In Section 3, we describe the design of TCP-ELSA. Section 4 compares the simulation results of TCP-ELSA with that of other protocols.

II. MOTIVATION

The combination of TCP congestion control and lossy wireless links causes severe degradation in the performance of TCP. Approaches to solve this problem span both the base stations and the end hosts, but it is unclear where the best solution lies. The base station is aware of losses caused by transmission errors, but it is not feasible to expect a base station to be optimized for TCP or even to be aware of the flows it is carrying. On the other hand, the end host is ultimately responsible for recovering from losses, but is not aware of the cause of the loss or the state of the wireless link. In this section, we discuss current solutions at the base station and the end host. The goal of our approach is to provide an end-to-end approach based on explicit knowledge of the cause of a loss and implicit knowledge of the state of the wireless link.

At the base station, local retransmission schemes[1-4] have been proposed to hide wireless losses from the end host so that the wireless link appears reliable. When the base station detects that a packet is lost on the wireless link, it retransmits the packet until the packet is successfully received. Simple retransmission schemes may result in large variation in the end-to-end delay. Such schemes also experience poor channel efficiency due to the repeated retransmission of packets going to low quality links that may block the transmission of other packets destined for hosts with higher quality links. This type of head-of-line blocking wastes large portions of the bandwidth in the wireless link. On the other hand, if no retransmission is done for the packets lost on the wireless link, the performance of TCP flows will be greatly decreased, and the throughput of flows that go to a low quality link will be much lower than that of flows going to a high quality link. This difference of service quality for different flows violates the fairness requirement that demands that every flow should be able to deliver the same amount of data to its receiver.

Several base station packet scheduling schemes [5-10] that are based on link quality has been proposed to solve the problems of bandwidth efficiency and fairness. Some of these scheduling algorithms give packets going to higher quality links higher priority [5-6] to improve efficiency; others [6-10] give flows on lower quality links more bandwidth to compensate for their high loss rate to equalize the goodput of each flow. Although these methods can greatly improve bandwidth efficiency and fairness for shared wireless links, they impose a heavy burden on the base station, which must keep per flow state and calculate schedules for all arriving packets. For a large wireless network, a base station may have hundreds of flows going through it and this burden may be too heavy to be practical.

At the endhost, approaches have been proposed which differentiate between losses due to packet corruption and losses due to congestion[11-15]. If a loss is caused by congestion, the TCP sender reduces its congestion window. If a loss is due to packet corruption, the TCP sender retransmits the lost packet and does not shrink its congestion window.

---

[1] This work was supported in part by the National Science Foundation under grant ANI 0081308

Although these protocols do not require TCP specific changes to the base station, they do not consider the link quality of different flows and hence suffer from low efficiency due to the repeated retransmission of lost packets on low quality links.

An ideal scheme should combine the benefits of both the link-quality-dependent scheduling approaches and end-to-end approaches, so that high bandwidth efficiency and fairness can be achieved without imposing a heavy burden on the base station. We notice that in many proposed end-to-end wireless TCP schemes [11, 12, 14-16], the end host can differentiate wireless losses from congestion losses. Therefore in TCP-ELSA, instead of putting the link-quality-dependent rate adjustment in the base station, we implement this functionality in the end host to relieve the burden of packet scheduling on the base station.

The design philosophy of TCP-ELSA is the same as for congestion control in TCP. In the Internet, instead of depending on routers to shape the traffic of each flow, every TCP sender takes the responsibility of congestion control through adjusting its congestion window. Through the cooperation of all the TCP senders, the routers can be optimized to the task of forwarding packets. Similarly, for a large wireless network, caching per-flow state and performing link state dependent scheduling for each packet in the base station may decrease the performance of a base station. TCP-ELSA is designed to make TCP senders cooperate with each other so that the tasks the base station needs to perform can be greatly simplified.

III. PROTOCOL DESIGN

The first goal of TCP-ELSA is to improve the efficiency of bandwidth usage by adapting the congestion control and loss recovery policies to the current state of the wireless link. When the link quality is high (i.e. losses are rare and random), TCP-ELSA is aggressive and immediately retransmits losses due to transmission errors on the wireless link. When the link quality is low (i.e. losses are bursty), TCP-ELSA backs off allowing other flows with better link quality to transmit.

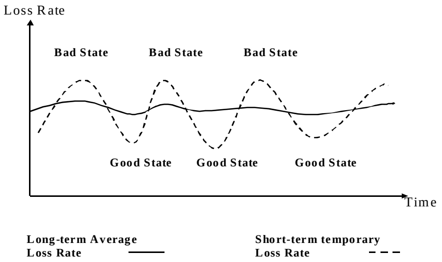


Figure 1: Good State and Bad State

The second goal of TCP-ELSA is to satisfy the basic fairness requirement of each flow. The long-term throughput of a flow should not depend on the quality of the link. It should only depend on the number of flows that share the link.

Fairness is achieved in TCP-ELSA because the rate of a flow depends on the short-term variation of link status instead of long-term link quality. Instead of defining the state of a link according to the absolute value of the loss rate, a link's state is defined in terms of the ratio of its temporary loss rate to its long-term average loss rate as depicted in Figure 1. The solid line in Figure 1 represents the long-term average loss rate. The dashed line represents the short-term temporary loss rate. If a link's temporary loss rate is higher than its average loss rate, the link is considered to be in a bad state; otherwise, it is considered to be in a good state.

It is expected that every link spend the same amount of time in its bad state and in its good state. Every flow experiences the same error free link in the good state because of the retransmission of lost packets; every flow backs off to let other flows in good states transmit when it is in bad state. Because each flow should approximately spend the same amount of time in good and bad state, the throughput of each flow should be approximately the same and fairness is achieved between flows.

In the rest of this section, we discuss how TCP-ELSA can distinguish between losses on the wired and wireless links and then use this information to govern its behavior.

A. *Loss Discrimination Techniques*

In order to adapt to the state of the wireless link, it is necessary for the sender to be able to differentiate between congestion-based and transmission-based losses. The most effective approach to providing such differentiation is the use of ELN (Explicit Loss Notification Message), where the base station informs the sender of any packets that could not be sent on the wireless link. Although such an approach does require modification to the base station, this is a simple management function and need not be aware of the number or type of flows being carried by the base station. The design of TCP-ELSA requires loss discrimination but does not depend on ELN. End-to-end techniques can be used to monitor the characteristics of a flow and approximate ELN through implicit information. This is current research topic and the effectiveness of such techniques is being evaluated.

B. *Reacting to Wireless Losses*

When the TCP sender is informed about packet losses on wireless link by ELN or other techniques, two facts have been revealed to it. First, the TCP sender learns which packets have been lost on wireless link and which on wired link. A TCP-ELSA sender only invokes congestion control for losses on the wired link. For losses on the wireless link, it retransmits the lost packets and resets the retransmission timer without cutting its window size. Second, the arrival rate of ELN messages along with the ACK indicates the packet loss rate on the wireless link. TCP-ELSA can adapt its sending rate, including retransmission rate of lost packets, according to the link state.

In order to know which lost packets to retransmit, each TCP sender maintains a loss history that records which packets are lost on the wireless link. Two parameters are kept per loss: the sequence number of the lost packet and a retransmission flag that indicates whether the lost packet has already been retransmitted. When a packet is retransmitted, its corresponding retransmission flag is set to true.

When a TCP sender gets an indication about a new packet lost on wireless link, the sender inserts a new entry into the loss history. The retransmission flag is set false for the new entry. The retransmission of the lost packet is based on the sending rate of TCP. If a retransmitted packet is lost again on wireless link, the TCP sender updates the retransmission flag of the corresponding entry to false. The rate of retransmission of lost packets depends on the state of the link.

TCP-ELSA estimates the link state by maintaining two parameters: the average packet loss rate (ALR) and the temporary packet loss rate (TLR), using a weighted moving average algorithm. By assigning different parameters to the weighted moving average, ALR can represents the long term quality of the wireless link and TLR can be set to be more sensitive to transient packet loss rate changes and indicate the current packet loss rate.

The sender adjusts its sending and retransmission rate according to ALR and TLR. When TLR is larger than ALR for a certain threshold, which signals that the wireless link is in a temporarily bad state, the sender goes into a rate-control state in which the sending rate slows down. In this state, TCP-ELSA does not resend the lost packet immediately upon receiving an ELN because the wireless link is in a bad state. Rather, a timer in the sender is set. When the timer expires, the sender chooses an entry in the loss history and retransmits the corresponding packet to probe the link. The probing frequency is related to the length of the bad state. If the average length of the bad state is large, the probing frequency should be low so that least bandwidth is used for probing. If the average length of the bad state is small, a higher probing frequency should be set so that a flow can get out of the rate-control state quickly.

If an ACK comes back from a probe message, the sender immediately retransmits another lost packet. After several rounds, TLR gets lower than ALR for a certain threshold, so the TCP sender gets out of the rate-control state and resumes its normal transmission. In this state, it resends the lost packet immediately after receiving an ELN and reset the retransmission timer.

C. *Handling Duplicate ACKs*

Besides the retransmission of the packets that are lost on the wireless link by ELN, TCP-ELSA must also correctly deal with duplicate ACKs. If the duplicate ACK is caused by a congestion loss, congestion control should be invoked; otherwise, the duplicate ACK has been caused by packet corruption in wireless link and there should be no congestion control in the TCP sender.

Our approach to distinguish between duplicate ACKs caused by losses on the wired link and losses on the wireless link is based on the sequence number of the lost packet. To simplify our analysis, assume the sequence number is increased by one for each subsequent TCP packet. If a packet is lost on a wireless link, the sender can learn the sequence number of the lost packet through the loss discrimination techniques such as ELN. When a duplicate ACK arrives that indicates a packet loss, the sender can tell whether the loss happened on wireless link or wired link. If the sequence number of the lost packet is in the loss history, the packet was lost on wireless link, otherwise the loss is a congestion loss.

IV. PERFORMANCE EVALUATION

In this section, we evaluate the performance of TCP-ELSA using simulations by comparing the bandwidth efficiency and fairness of two other TCP protocols with that of TCP-ELSA.

All simulation results are based on the Network Simulator (NS)[17]. We choose 802.11 as the MAC layer protocol on the wireless link.

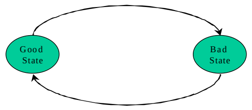
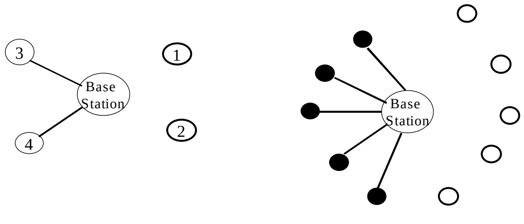


Figure 2: the Error Model

For all simulations, we use an error model with two states (as shown in Figure 2). The two states represent a low error rate state and a high error rate state. The transitions from good-to-bad state and from bad-to-good state are uniformly distributed between [3.2, 4.8] seconds. We also tried other error models, such as a two state Markovian model[18, 19] and observed similar results. Due to space limitations, we do not present the simulation results of other error models.



(a) (b)

The delay of the wired link is 40 ms. The mac layer protocol is 802.11

Figure 3: the Network Topology of Simulations

The simulation results in this section compare the performance of three protocols: TCP New Reno, TCP EBSN and TCP-ELSA. In TCP-EBSN [15], the base station sends a ELN message, which is called Explicit Bad State Notification (EBSN) in [15], to the TCP sender when a packet is lost and retransmits the packet until it is successfully delivered to the sender. Upon receiving the EBSN, the TCP sender reset its retransmission timer so that it will not timeout and retransmit the packets that are being retransmitted in the base station.

A. Fairness in Wireless Link Sharing

In this simulation, we compare the fairness behavior of TCP New Reno, TCP-EBSN and TCP-ELSA.

The network configuration for the simulation is shown in Figure 3(a). There are two wired nodes that each establishes a TCP connection with a wireless node. The bandwidth of the wired links is 5Mbps and the maximum bandwidth of the wired link is 2Mbps. The bit error rate for the link to wireless Node 1 is uniformly distributed in $[10^{-4}, 10^{-2}]$ in bad state and $[10^{-7},10^{-5}]$ in good state. The bit error rate for the link to wireless Node 2 is uniformly distributed in $[10^{-3}, 10^{-1}]$ in bad state and $[10^{-6},10^{-4}]$ in good state. The average quality of the link to Node 1 is 10 times better than the average quality of the link to Node 2. The first TCP connection is established between Nodes 4 and 2 at time 2 seconds. The second TCP connection is established from Node 3 to Node 1 at time 20 seconds. The simulation runs 100 seconds.

Figure 4 depicts the trace of packets of the two connections using TCP New Reno. Flow 1, which is the connection between Nodes 4 and 2, experiences a very low throughput due to the high loss rate on the link between the base station and node 2. On the other hand, Flow 2, which goes from Node 3 to Node 1, gets high throughput due to its relatively low loss rate. It is obvious that TCP is very unfair to the flow that experiences high packet loss rate on the wireless link.

Figure 5 shows the trace of packets using TCP-ESBN. Because TCP-ESBN uses local retransmission to hide wireless losses from the TCP sender, the throughput of each connection is not related to the wireless loss rate. Both TCP-ESBN connections get the same amount of bandwidth from the base station. So fairness is achieved in TCP-ESBN. Also because of local retransmissions, the combined throughput of both TCP-ESBN connections is higher than that of TCP New Reno connections.

Figure 6 demonstrates that TCP-ELSA not only achieves a higher combined throughput than that of TCP-EBSN, but also meets the fairness requirement. In this scenario, both of the TCP-ELSA flows have a higher rate than that of the corresponding TCP-EBSN flows while at the same time share the wireless link equally. The probing frequency of TCP-ELSA is set as 1 second and simulation result shows that this frequency is high enough for the 4 second average duration of bad periods.

B. Bandwidth Efficiency

In this simulation, we demonstrate that TCP-ELSA can greatly improve the bandwidth usage efficiency in the wireless channel.

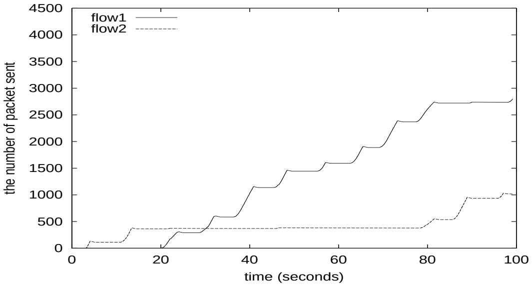


Figure 4: TCP Packet Trace of Two Competing Flows

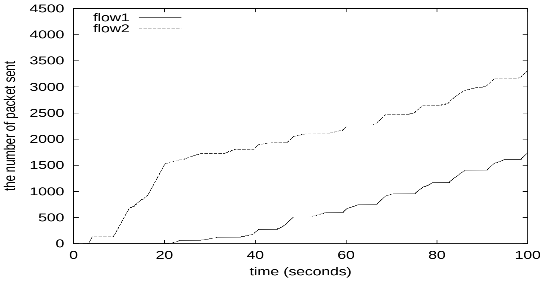


Figure 5: TCP-EBSN Packet Trace of Two Competing Flows

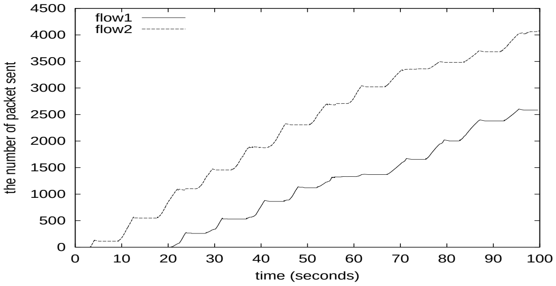


Figure 6: TCP-ELSA Packet Trace of Two Competing Flows

Figure 2 (b) shows the network topology. In the wireless links, the average bit error rate in bad state is 1000 times larger than the average bit error rate in good state. For each state, the distribution of bit error rate is uniform on [average bit error rate/2, average bit error rate*2]. Each of the five wired nodes establish a TCP connection with one of the wireless nodes at the beginning of the simulation. Each simulation runs 100 seconds.

Figure 7 shows the total throughput of the five flows at different bit error rates. The horizontal axis shows the average bit error rate of the good state. The vertical axis shows the total throughput of the five TCP flows.

As shown in Figure 7, when the average bit error rate is low, TCP New Reno, TCP-EBSN and TCP-ELSA all show high throughput because loss is random and rare. As the average bit error rate increases, the throughput of all three protocols decreases. TCP-EBSN's performance is worst when the bit error rate in the good state is between $[10^{-6}, 2.5*10^{-4}]$. This is because as the packet loss rate in the bad period increases, the local retransmission of lost packets in the base station may repeatedly fail so that a large portion of the wireless bandwidth is wasted on retransmitting lost packets to links that are in a bad state. Interestingly, TCP

New Reno's performance is better than TCP-ESBN in this range of bit error rates. This can be explained in the packet trace in Figure 8. During the time between [43, 48] seconds, the link is in a bad state, the sender of TCP New Reno times out and invokes the congestion control algorithm to reduce its sending rate. Hence, the flows that experience a good link can enjoy more bandwidth on the wireless link because the flows in the bad state back off. The total throughput of the five flows is better than TCP-EBSN because the congestion control mechanism of TCP happened to always let the flows that experience the good link transmit.

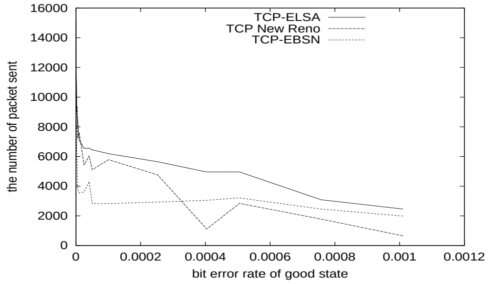


Figure 7: The throughput of TCP, TCP-EBSN and TCP-ELSA under different bit error rate (five competing flows)

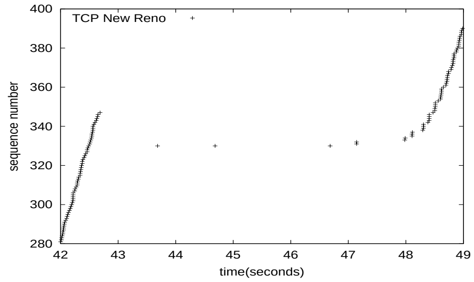


Figure 8: The detailed packet trace of TCP (five competing flows)

When the average bit error rate in good state is over $2.5*10^{-4}$, the performance of TCP New Reno becomes worse than TCP-EBSN. This is because in this range of bit error rates, the wireless losses in a good period are not rare anymore. Without local retransmission, the TCP New Reno sender backs off on wireless losses even when the link is in a good state so that the total throughput is greatly reduced.

Note that, TCP-ELSA's throughput is always higher than both of TCP-EBSN and TCP New Reno because it neither blindly retransmits the lost packet as TCP-EBSN nor blindly backs off on losses as TCP New Reno. TCP-ELSA adjusts its behavior according to the link quality. If the link quality is in a relatively good state, TCP-ELSA retransmits lost packets immediately; if the link quality is in a relatively bad state, TCP-ELSA backs off to let other flows that are in a good state use the channel.

V. CONCLUSION

This paper presents a study of the effect of burst errors on competing TCP flows. Maintaining high channel utilization and ensuring fair distribution of bandwidth to different connections are desirable in transporting TCP traffic over wireless links. We have demonstrated that both TCP and local retransmission protocols cannot achieve these two goals at the same time and their performance is not good for all ranges of bit error rates. We have described how a wireless loss aware TCP sender can adjust its sending rate to achieve both the fairness and efficiency goals and maintain a good performance in a large range of bit error rates.

REFERENCES

[1] Balakrishnan, H., et al. Improving TCP/IP Performance over Wireless Networks. in 1st ACM International Conference on Mobile Computing and Networking(Mobicom).November, 1995.

[2] Mehta, M. and N.H. Vaidya, Delayed duplicate acknowledgements: A proposal to improve performance of TCP on wireless links. 1998, Texas A&M University.

[3] Mehta, M.N. and N.H. Vaidya, *Delayed Duplicate Acknowledgements: A TCP-unaware Approach to Improve Performance of TCP over Wireless*. 1999, Computer Science Department, Texas A&M University.

[4] Chochalingam, A. and M. Zorzi, *Wireless TCP Performance with Link Layer FEC/ARQ*. 1999.

[5] Bhagwat, P., et al. *Using channel state dependent packet scheduling to improve throughput over wireless channels*. in *IEEE INFOCOM*.March, 1996.

[6] Fragouli, C., V. Sivaraman, and M.B. Srivastava. *Controlled Multimedia Wireless Link Sharing via Enhanced Class-Based Queuing with Channel-State-Dependent Packet Scheduling*. in *IEEE INFOCOM*.March, 1998.

[7] Lin, P., et al. *A wirless fair scheduling algorithm for error-rone wireless channels*. in *Proceedings of ACM WoWMOM*2000.

[8] Lu, S., V. Bharghavan, and R. Srikant. *Fair scheduling in wireless packet networks*. in *Proceedings of ACM SIGCOMM*1997.

[9] Lu, S., T. Nandagopal, and V. Bharghavan, *A wireless fair scheduling algorithm for error-prone wireless channels*. Wireless Networks Journal, February, 1999.

[10] Ng, T.S.E., I. Stoica, and H. Zhang. *Packet fair queueing algorithms for wireless networks with location-dependent errors*. in *Proceedings of IEEE INFOCOMM*1998.

[11] Balakrishnan, H., S. Seshan, and R. Katz. *Explicit loss notification and wireless web performance*. in *Proceedings of the IEEE Globecom Internet Mini-Conference*.November, 1998.

[12] Chen, W.-P., et al., *Syndrome: A light-weight approach to improving TCP performance in mobile wireless networks*. The journal of Wireless communications and Mobile computing, April, 2001.

[13] Cobb, J.A. and P. Agrawal. *Congestion or corruption? A strategy for efficient wireless TCP sessions*. in *Proceedings of Computer and Communications*1995.

[14] Floyd, S., *TCP and Explicit Congestion Notification*. ACM Computer Communication Review, Oct., 1994. **24**(5).

[15] Bakshi, B.S., et al. *Improving performance of TCP over wireless networks*. in *17th International conference on distributed computing systems*.may, 1997.

[16] Goel, S. and D. Sanghi. *Improving performance of TCP over wireless links*. in *Proceedings of TENCON 98 - 1998 IEEE Region Ten Conference on Global Connectivity in Energy, Computer Communication and Control*.December 17-19, 1998.

[17] Fall, K. and K. Varadhan. *ns notes and documentation*. in *The VINT Project*.November, 1997. UC Berkely, LBL, USC/ISI, and Xerox PARC.

[18] Swarts, E. and H. Ferreira, *Markov characterization of digital fading mobile VHF channels*. IEEE Trans. Behic. Tech, November, 1994: p. 977-985.

[19] H.Wang and N.Moayeri, *Finite state markov channel-a useful model for radio communication channels*. IEEE Trans. Vehic. Tech., Feburary, 1995: p. 163-171.