

基于 TCP 友好的无线网络拥塞控制机制研究

肖 甫 王汝传 孙力娟 王华顺

(南京邮电大学计算机学院 南京 210003)

摘 要 网络实时多媒体业务的广泛应用对传统传输层协议提出了新的挑战:拥塞控制机制的缺乏使得 UDP 严重抢占 TCP 应用的共享带宽,从而降低网络的公平性,甚至导致网络拥塞。针对无线网络的高误码特性,将传输延时抖动引入到 TFRC 控制机制中,提出了一种基于速率控制的 TCP 友好拥塞控制算法 TFRC-JI。该算法基于传输延时抖动有效区分无线链路的拥塞和误码,并以此反馈至发送端,实现不同的速率控制机制。实验结果表明,与传统的 TFRC 相比,改进的 TFRC-JI 在保持对 TCP 业务友好性的同时实现了链路的高效使用,并降低了传输时延抖动,从而较好地适应多协议共存的无线网络实时业务传输。

关键词 无线网络,拥塞控制,TFRC,高误码

中图分类号 TP393.03 **文献标识码** A

Research of TCP-friendly Congestion Control Protocol in Wireless Network

XIAO Fu WANG Ru-chuan SUN Li-juan WANG Hua-shun

(School of Computer Science and Technology, Nanjing University of Posts and Telecommunications, Nanjing 210003, China)

Abstract The network real-time multimedia service's widespread application presented an new challenge to the traditional transport protocol; the deficiency of congestion control mechanism causes UDP to seriously seize the sharing band width of the TCP application, thus reduces the network fairness, even causes network congestion. Based on the characteristic of high error rate in wireless network, a novel TCP friendly congestion control algorithm TFRC-JI was presented in this article, which introduces the latency vibration to distinguish the link congestion from code error, thus different speed control mechanism is feedback to the transmitting end. Simulation experiment result indicated that compared with the traditional TFRC mechanism, the throughput of wireless link is enhanced while TCP friendly also maintains using TFRC-JI, which suites well for real-time service transmission.

Keywords Wireless network, Congestion control, TFRC, High error rate

1 引言

随着 Internet 技术的不断发展,人们对网络的应用也从传统的 WWW、E-mail 到网络电视、网络电话、视频会议等多媒体应用,多样化的应用业务流,尤其是多媒体流占据了 Internet 整体流量相当大的部分,而巨大的流媒体数据量很容易引起网络过载,造成网络拥塞甚至崩溃,这对网络拥塞控制技术又提出了新的挑战。

拥塞控制是确保 Internet 鲁棒性的一个关键因素,拥塞控制机制的研究是学术界的热点问题之一。引起链路拥塞的原因是多方面的,但究其根本原因在于供给不平衡,即网络所提供的带宽不足以满足人们对网络负载的需求,对此最直接的解决办法是增加带宽。然而,仅仅依靠提高网络带宽并不能完全解决拥塞问题,应用的带宽需求的增长速度仍远大于

带宽的增长速度。并且,由于 Internet 的异构性和负载的不确定性,流量在时间、空间上的分布不均,也会导致局部或者暂时的拥塞^[1]。目前,Internet 上的业务流主要是基于 TCP 和 UDP 这两种基本传输协议。UDP 作为一种无连接的传输协议,比 TCP 更适合媒体流。然而,UDP 没有拥塞控制机制,当基于 UDP 的应用和基于 TCP 的应用共享网络带宽时,前者将获得大量带宽,而 TCP 流因慢启动、拥塞避免、快速重传和快速恢复等拥塞机制而不公平地分配带宽,可能更进一步导致拥塞崩溃。友好的拥塞控制是网络传输的一个重要机制。研究及实践表明,如果缺乏有效的友好拥塞控制机制,会导致链路拥塞甚至崩溃等严重后果。因此如何实现 Internet 中友好的拥塞控制机制,对于多种数据流共存网络环境下进行性能研究和分析具有重要意义。

本文第 2 节综述了地面标准 TCP 友好拥塞控制机制;第

到稿日期:2009-08-12 返修日期:2009-10-30 本文受国家自然科学基金(60973139,60773041,60903181),中国博士后科学基金(20090451240,20090451241),2006 江苏省软件专项,江苏省博士后科研资助计划(0801019C),江苏省六大人才高峰项目(2008118)和南京市高科技项目(2007 软资 106)资助。

肖 甫(1980-),男,博士后,副教授,硕士生导师,CCF 会员,主要研究方向为无线传感器网络、卫星网络、多媒体技术等,E-mail: xiaof@njupt.edu.cn;王汝传(1943-),男,教授,博士生导师,CCF 会员,主要研究方向为无线传感器网络、信息安全、普适计算等;孙力娟(1963-),女,博士,教授,博士生导师,主要研究方向为无线网络协议与智能算法;王华顺(1982-),男,硕士生,主要研究方向为基于通信网络的计算机应用技术。

3 节在回顾传统 TFRC 基本原理的基础上,针对无线网络的高误码特性提出了一种改进的 TFRC 机制 TFRC-JI;第 4 节对 TCP,UDP,TFRC 及改进版本与本文的 TFRC-JI 进行了仿真实验及分析比较;最后总结全文。

2 TCP 友好拥塞控制机制

TCP 友好拥塞控制是在上世纪 90 年代提出的,经过近 20 年的发展,已经提出了不少 TCP 友好拥塞控制机制。按照调整网络负载的方式可将 TCP 友好拥塞控制机制分为基于窗口和基于速率两大类^[2]。基于窗口的 TCP 友好拥塞控制机制是在发送方或接收方使用一个拥塞窗口来实现 TCP 友好,这与 TCP 协议的拥塞控制机制类似,典型代表有 GAIND,BCCA 等^[3,4]。但其采用基于窗口的拥塞控制机制,可能会产生突发流,造成较大的抖动,因此不太适合流媒体应用。基于速率的 TCP 友好拥塞控制机制根据网络中指示拥塞的信息来调整传输速率,以达到 TCP 友好。该类友好拥塞控制机制是当前的一个重要研究方向,典型算法包括 RAP^[5],LDA+^[6],TEAR^[7],TFRC^[8]等。RAP 采用适配算法较好地降低了传输抖动,但是每包反馈的机制加重了网络的负担。TEAR 采用在接收端根据速率模型计算发送速率并将该速率发送到发送端,来实现速率控制。由于没有采用 RAP 那样的每包确认机制,因此较好地适应了组播和媒体流的应用,但是该方案实现机制比较复杂,会给接收端带来较大的负担。LDA 协议则利用 RTP 进行传输,其主要优点是在减小发送速率振动的同时也抑制了流量突发;其主要缺点则是一旦丢包,就减小发送速率,可能会带来较大抖动,降低媒体流的平稳性^[9]。文献[10]采用延时抖动对 TFRC 进行了改进,但其主要出发点是降低实时业务的传输单向延时抖动,没有考虑拥塞和误码,对于大延时抖动情况只是减慢发送速度,不利于网络带宽的高效利用。

3 面向无线网络的改进 TFRC 协议

3.1 TFRC 协议基本原理

TFRC 作为当前研究最深入及使用最成熟的基于速率控制的 TCP 友好传输协议,在防止网络拥塞崩溃的同时充分利用瓶颈链路带宽,对 TCP 流公平而且平滑地进行速率调节。TFRC 既可以实现与 TCP 流公平地共享带宽,又不会因为单个包的丢失而将速率减半造成抖动,能够保持相对稳定的发送速率,比较适合基于 UDP 的多媒体流传输的应用。在具体算法实现中,由于需要对 TCP 保持友好性,TFRC 使用的是在 TCP-Reno 协议中的吞吐率计算公式^[11],即

$$T = \frac{S}{R\sqrt{\frac{2bP}{3}} + t_RTO\sqrt{\frac{3bP}{8}}P(1+32P^2)}$$

式中, T 为吞吐率(单位:B/s); S 为数据包的大小(单位:B); R 为往返延时 RTT(单位:s); P 为稳定状态下的丢包事件率,它是在一个发送窗口内发生的丢包事件数与发包总数的比值; t_RTO 为重传时钟(单位:s),一般直接设置为 $4 * RTT$,这种粗粒度的 RTO 计算公式被证明是简单有效的。 b 是单个 TCP ACK 表示接收的报文数,比如接收端收到 2 个数据包,便发送 1 个 ACK,则 b 设置为 2;由于很多实现没有采用延迟的 ACK,因此建议 RFC 设置为 $b=1$ 。TFRC 的基本实现过程是在接收端检测网络的拥塞信号,这里利用的是

丢包率并将信号反馈给发送端。而发送端利用以丢包率和回环时间为变量的吞吐率公式来调整发送速率,以适应网络拥塞的情况。发送端若没有收到反馈,也会减小发送速率。TFRC 协议有很好的稳定性,同时它并不是发现丢包就直接减半发送速率,而是在持续发现几个丢包事件后开始减半的,因此适合多媒体流应用。

3.2 改进的拥塞控制机制 TFRC-JI

之初提出的 TFRC 是面向有线地面网络的,但将其直接应用于当前广泛使用的无线网络时,性能则会大大下降。主要原因在于有线网络中丢包可以作为网络拥塞的信号,而无线链路自身误码率较高,因此丢包可能是链路误码引起的。传统 TFRC 无法区分丢包的具体原因,只要检测到丢包就通过速率模型降低发送方速率。此时,若丢包确实是由网络拥塞引起的,则这种调节是正确的;而如果丢包是误码引起的,则对拥塞控制情况作出了错误的判断,在网络并不拥塞的情况下毫无必要地降低了发送方速率,导致网络吞吐量下降。针对以上不足,我们对传统的 TFRC 进行改进,提出了一种新的 TCP 友好拥塞控制协议 TFRC-JI。TFRC-JI 主要是对传统 TFRC 的平均丢包事件率 P 的计算方法进行改进,从而使其能够较好地区分具体的丢包原因从而采取不同的调节策略。

定义 1 平均丢包事件率 p 为一定时间记录内平均丢包间隔的倒数。

$$P = \sum_{i=0}^n w_i / \sum_{i=0}^n A_i w_i$$

式中, A_i 为最近的第 i 个丢包间隔; W_i 为对应的权重,我们取其一维高斯核函数,即

$$W_i = e^{-\frac{t}{T}}$$

延迟抖动为某连接中两个相邻数据包的传输延迟差值。设 S_i 为数据包 i 的发送时间, R_i 为数据包 i 的到达时间,则对于相继两个数据包 i 和 $i+1$,延迟抖动可表示为

$$D(i, i+1) = (R_{i+1} - S_{i+1}) - (R_i - S_i)$$

从延迟抖动公式中可以看出,当 $D=0$ 时,数据包 i 和 $i+1$ 在传输中的延迟相等;若 $D>0$,则表示数据包 $i+1$ 传输的时间要长于数据包 i ,即认为 $i+1$ 在网络中被排队处理。因此,延迟抖动可以作为丢包原因的一种区分:当抖动大于某个设定的阈值时,可认为当前网络处于拥塞状态,对应丢包将被认为是拥塞丢包;而当数据包延时抖动低于设定阈值时,则认为此时丢包为误码丢包。

TFRC-JI 引入了新的统计包间隔的方法,即以当前的延时抖动是否超过阈值来判断丢包的原因,并据此对当前的丢包间隔进行调整。当延时抖动大于设定阈值 K 时,则认为网络处于拥塞状态,此时按一定的比例因子减小当前统计的丢包间隔,进而增大丢包事件率 P 。而 P 的增大使得发送方降低其发送速率,从而达到拥塞控制的目的;当延时抖动小于设定阈值 K 时,则认为网络处于非拥塞状态,当前丢包被认为是误码丢包,按照一定的比例因子扩大当前的丢包间隔,以增大当前发送方的发送速率,提高网络吞吐量。

根据 TCP VenO 方程^[12],标准的调整因子应取 3,由此引入的调整因子函数应是以 3 为分隔点的分段函数。当延时抖动大于预先设定的阈值 K 时,即网络处于拥塞状态,调整因子函数应该介于 $[0,1]$ 之间,以降低丢包间隔,从而增大丢包

事件率,降低发送速度;反之,调整因子函数就应该介于[1,3]之间,以扩大丢包间隔,增大发送速度。

调整因子对应函数 $A(t)$ 定义为

$$A(t)=\begin{cases} e^{-(t-K)^2/(2\sigma^2)}, & t>K \\ -2t/K+3, & t\leq K \end{cases}$$

$A(t)$ 介于[0,1]之间时,用于减小当前丢包间隔; $A(t)$ 介于[1,3]之间时,用于增大当前丢包间隔。

传统 TFRC 协议用于无线网络时,由于不能准确区分误码与拥塞丢包,从而造成不必要的发送速率降低,使得网络吞吐量下降。对此,我们提出的 TFRC-JI 通过在接收端以单向传输延迟抖动为信号来区分丢包的具体原因,从而加强对单向传输延时稳定性的控制,大大降低系统的抖动性,使之更利于传输端到端实时通信业务,最终在保证 TCP 友好的前提下,提高无线环境下网络的吞吐量。

4 实验及分析

基于 NS2 仿真软件对 UDP,TFRC 和 TFRC-JI 在无线环境下的 TCP 友好性以及网络的整体性能进行了验证和分析。仿真的拓扑结构如图 1 所示,其中 n2-n3 链路为瓶颈链路,带宽为 1.7Mb,延时为 30ms;非瓶颈链路 n0-n2,n1-n2 带宽为 2Mb,延时为 20ms,队列丢包方式都采用 Droptail 方式。节点 n0-n3 上绑定 TCP 代理,在仿真过程中固定进行 TCP 流的传输,而节点 n1-n3 之间则根据仿真需要分别绑定 UDP,TFRC 以及本文所提出的 TFRC-JI 协议,并考察分析不同协议情况下对 TCP 的友好性以及网络的整体性能。

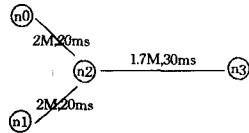


图 1 仿真网络拓扑

首先考察基于流媒体的 UDP 与 TCP 共存的情况。在上述拓扑结构中,n0 节点绑定 TCP 代理,n1 节点绑定 UDP 代理,其中 TCP 流在 0.1s 开始发送,而 UDP 流在 5s 时开始发送,仿真时间为 100s。仿真吞吐量结果如图 2 所示。

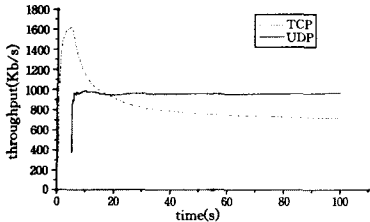


图 2 TCP/UDP 环境下网络吞吐性能比较

从图 2 可以看出,UDP 流在第 5s 开始启动后,快速占据带宽且基本保持在 1Mb 左右,具有较好的平滑特性,从而适用于流媒体传输。但友好性的缺乏使其对带宽争夺、对 TCP 流造成了极大的攻击性,甚至会导致 TCP 流饿死,造成网络崩溃。

在完全相同的网络配置下,分别将 TFRC,TFRC-JI 代理绑定到 n1 节点进行相同的仿真实验,结果如图 3、图 4 所示。对比图 2,图 3 可以发现,TFRC 的引入在满足基于 UDP 的上层网络应用的前提下很大程度上缓解了 UDP 流对 TCP 流

的不友好性。图 2 中 TCP 流的平均吞吐率为 893kb/s,而引入 TFRC 后 TCP 流的平均吞吐率增加到 1041kb/s 左右。而且 TFRC 由于是基于速率模型,因此很好地保持了 UDP 流的平滑特性。

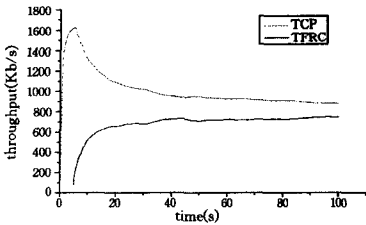


图 3 TCP/TFRC 环境下网络吞吐率比较

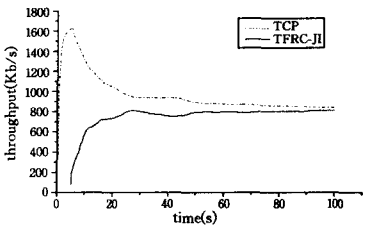


图 4 TCP/TFRC-JI 环境下网络吞吐率比较

将 TFRC-JI 替代 TFRC,重新进行实验,结果如图 4 所示。可以发现,TFRC-JI 不仅保持了对 TCP 流的友好性,而且较大提高了基于 TFRC 流的吞吐量以及整个瓶颈链路的吞吐量。在 TFRC 流、TCP 流共存网络中,TCP 流的平均吞吐率为 1040kb/s,TFRC 流的平均吞吐量为 690kb/s;而将 TFRC-JI 代替 TFRC 后,TCP 流的吞吐率约为 1003kb/s,TFRC-JI 流平均吞吐率则为 760kb/s。TFRC-JI 相对于 TFRC 吞吐率提高了 70kb/s,而相对的 TCP 流仅降低了 37kb/s,由此在提高网络整体吞吐量的同时保持了对 TCP 流的友好性。

图 5 为分别使用本文改进的 TFRC-JI 和传统 TFRC 算法时网络中非面向连接应用的吞吐性能的比较。可以看出,本文改进算法具有较好的性能。

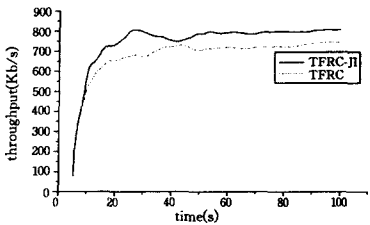


图 5 TFRC-JI/TFRC 吞吐率性能比较

在以上环境中分别用 TFRC,TFRC-CJUTD^[10] 以及 TFRC-JI 算法进行同样的仿真,网络参数配置同上,结果(吞吐率、传输延时抖动)如表 1 所列。

表 1 TFRC,TFRC-CJUTD 和 TFRC-JI 算法结果比较

算法	TFRC 吞吐率	TCP 吞吐率	总吞吐率	平均延时抖动
TFRC	692.9	1041.4	1734.3	0.0110
TFRC-CJUTD	728.0	1019.1	1747.1	0.0106
TFRC-JI	763.4	1003.6	1767.0	0.0099

从表 1 中可以看出,应用 TFRC 算法时,TFRC 流共存的 TCP 流的吞吐率比值为 0.665;相应地在应用 TFRC-CJUTD 算法时,TFRC-CJUTD 流与共存的 TCP 流的吞吐率比值为 0.714;而应用 TFRC-JI 算法时,TFRC-JI 流与共存的 TCP 流

的吞吐率比值为 0.761。可以看出, TFRC 较好地保持了对于 TCP 业务流的友好性。在应用 TFRC 算法时, TFRC 流的平均延时抖动为 0.0110ms; 在应用 TFRC-CJUTD 算法时, TFRC-CJUTD 流的平均延时抖动为 0.0106ms; 在应用 TFRC-JI 算法时, TFRC-JI 流的平均延时抖动为 0.0099ms, 相对 TFRC 以及 TFRC-CJUTD 分别下降了 10% 和 6.6%。以上数据表明, 和已有的 TFRC 及改进算法比较, TFRC-JI 在保持对 TCP 业务友好性的同时实现了链路的高效使用, 并降低了传输时延抖动, 从而较好地适应多协议共存的无线网络实时业务传输。

结束语 传统基于 UDP 的应用尤其是多媒体应用对带宽资源争用严重, 这使得它在与 TCP 流共存的网络中表现出了对 TCP 极大的攻击性。本文通过深入分析典型的 TCP 友好拥塞控制协议 TFRC 及其在无线环境下的不足, 提出了一种改进的 TFRC-JI 机制, 它通过传输延时抖动有效区分无线链路的拥塞和误码, 并以此反馈至发送端, 实现不同的速率控制机制。仿真实验验证了 TFRC-JI 在保持对 TCP 业务友好性的同时实现了链路的高效使用, 并降低了实时业务流的传输时延抖动, 从而较好地适应多协议共存的无线网络实时业务传输。

参考文献

- [1] 刘绍辉. Internet 拥塞控制相关算法研究及仿真分析[D]. 成都: 西南交通大学, 2005
- [2] Widmer J, Denda R, Mauve M. A Survey on TCP-friendly Con-

gestion Control[J]. IEEE Network Magazine, 2001, 15(3): 28-37

- [3] Yang Y, Lam S. General AIMD Congestion Control[C]// Proceedings of ICNP 2000. Osaka, Japan, Nov. 2000
- [4] Bansal D, Balakrishnan H. Binomial Congestion Control Algorithms[C]// Proceeding of IEEE INFOCOM 2001. Anchorage, Alaska, Apr. 2001: 631-640
- [5] Rejaie R, Handly M, Estrin D. RAP: An End-to-End Rate-Based Congestion Control Mechanism for Realtime Streams in the Internet[C]// Proceedings of IEEE INFOCOM99. Mar. 1999
- [6] Sisalem D, Wolisz A. LDA+: TCP-friendly Adaptation Scheme for Multimedia Communication[C]// Proceedings of NOSS-DAV'98. Cambridge, England, July 1998
- [7] Rhee I, Ozdemir V, Yi Y. TEAR: TCP emulation at receivers-flow control for multimediasstreaming[R]. Dept. of Comp. Sci, NCSU, Apr. 2000
- [8] Handley M, Floyd S, Padhye J, et al. TCP Friendly Rate Control (TFRC)[S]. Protocol Specification, RFC3448. IETF, 2003
- [9] 吴东. TCP 友好拥塞控制研究[D]. 南宁: 广西师范大学, 2006
- [10] 甘泉, 薛质. 控制端到端传输延迟抖动的改进 TFRC 算法[J]. 计算机工程, 2008, 34(10): 105-107
- [11] Wen P, Cao J, Li Y. Design of High-performance Networked Real time Control Systems [J]. Control Theory & Applications, 2007, 1(5): 1329-1335
- [12] Zhou B, Fu C P, Chiu D M, et al. A Simple Throughput Model for TCP Veno[C]// IEEE ICC 06. June 2006

(上接第 49 页)

考虑全局信息, 在时间复杂度上有一定优势。同时, 这种局部社团划分算法也可以应用于全局网络的社团结构的搜索, 并且不需要事先了解社团规模或者全局社团个数。把这种算法应用在 Zachary 俱乐部网络进行实验, 取得了较好的划分效果。该算法对 JDK 的 JSCG 网络图的划分结果也证实了 JDK 提供的功能模块的特点。但是该算法在选择节点时都是选择局部最优值, 可能最终产生的解不是全局最优解, 下一步工作将考虑结合模拟退火算法对其进行改进。

在软件开发设计过程中, 需要将软件划分成不同模块进行开发。在 Java 软件开发设计中, 当确定了类、接口等之间的相互关系后, 可以对生成的 JSCG 图利用本算法进行社团划分从而形成开发的模块, 以便于软件设计和开发。下一步工作也将应用此算法对其它软件进行社团划分和分析。

参考文献

- [1] Watts D J, Strogatz S H. Collective dynamics of 'small world' networks[J]. Nature, 1998, 393: 440-442
- [2] Barabási A L, Albert R. Emergence of scaling in random networks[J]. Science, 1999, 286: 509-512
- [3] Newman M E J. Detecting community structure in networks[J]. J. Eur Phys J B, 2004, 38: 321-330
- [4] Girvan M, Newman M E J. Community structure in social and biological networks[J]. J. Proc. Natl. Acad. Sci. USA, 2002, 99:

7821-7826

- [5] Radicchi F, Castellano C, Cecconi F F, et al. Defining and identifying communities in networks[J]. Proc Natl Acad Sci USA, 2004, 101: 2658-2663
- [6] Newman M E J, Girvan M. Finding and evaluating community structure in networks[J]. J. Phys Rev E, 2004, 69: 026113
- [7] Newman M E J. Fast algorithm for detecting community structure in networks[J]. Phys Rev E, 2004, 69: 066133
- [8] Clauset A. Finding local community structure in networks[J]. J. Phys Rev E, 2005, 72: 026132
- [9] Bagrew J P, Bolit E M. A Local Method for Detecting Communities[J]. J. Phys Rev E, 2005, 72: 046108
- [10] Girvan M, Newman M E J. Community structure in social and biological networks[J]. J. Proc Natl Acad Sci USA, 2002, 99: 7821-7826
- [11] 解治, 汪小帆. 复杂网络的一种快速局部社团划分算法[J]. 计算机仿真, 2007, 24(11): 82-85
- [12] 陈焘, 顾庆, 王树森, 等. 基于复杂网络的 JDK 代码结构演化研究[J]. 电子学报, 2007, 35: 118-123
- [13] Maqbool O, Babri H A. Hierarchical Clustering for Software Architecture Recovery[J]. IEEE Trans. on Software Eng, 2007, 33(11)
- [14] 杨博, 刘大有, Liu Jiming, 等. 复杂网络聚类方法[J]. 软件学报, 2009, 20: 54-66