Scholar Publishing Group
International Journal of Big Data Intelligent Technology
https://doi.org/10.38007/IJBDIT.2026.070204
ISSN 2790-0932 Vol. 7, Issue 2: 31-39

Android Communication Reliability Enhancement Architecture Based on Multi Link Diversity Transmission

Junhao Su

Jacobs school of engineering, University of California San Diego, La Jolla, CA, 92093

Keywords: Android; Multipath Diversity; Multipath QUIC; Communication Reliability; Cross-Layer Scheduling

Abstract: To address the vulnerability of Android terminals in mobile scenarios to single-link jitter, handover interruptions, weak coverage, and transient congestion, which can lead to message retransmissions, session interruptions, and sudden increases in latency, this paper proposes a multi-link diversity transmission reliability enhancement structure characterized by collaborative access from WiFi and cellular networks. A four-layer closed loop of "perception, decision-making, scheduling, and recovery" is created on the device side. Link quality estimation is used to prioritize packets, and redundant transmission and rapid failure rescheduling are implemented to provide differentiated protection for control messages and continuous service flows. Based on recent research on Multipath QUIC, partially reliable transmission, and mobile multi-connectivity, middleware suitable for reliability enhancement between the Android application layer and transport layer is designed, and session continuity criteria, link availability models, weighted scheduling strategies, and redundancy triggering mechanisms are presented. Comparison with real-world experimental results from published literature and Android mobile network datasets shows that combining multi-link diversity and partially reliable transmission can significantly reduce tail latency, buffer interruption time, and retransmission pressure. Research shows that improving the reliability of the Android system in complex wireless environments cannot rely solely on improving a single protocol. Instead, it should be approached from multiple angles, relying on the cooperation of various interfaces, awareness between upper and lower layers, and service level classification to achieve systematic improvement.

1. Introduction

With the increasing applications of mobile office, vehicle network control, real-time audio and video, instant messaging, and industrial mobile terminals, Android devices have transformed from consumer access terminals into critical edge nodes that can remain online for extended periods without high interference and with high reliability. Especially in subways, campuses, hospitals, factories, and urban scenarios, users' terminals often need to constantly switch between WiFi and 4G/5G cellular networks. Single-link communication is affected by various factors such as coverage changes, network congestion, inter-site handover, and radio frequency interference, resulting in situations such as short-term session disconnection, duplicate message submission, service lag, and increased tail latency. Recent experimental and research results show that Multipath TCP, Multipath QUIC, partially reliable transmission, connection migration, and multi-connection structures can enable mobile terminals to utilize multiple heterogeneous network paths to improve throughput, smooth switching, and maintain service continuity to some extent [1-4].

However, most existing solutions rely on video streams, general transmission protocols, or network emulation, while systematic architectures directly addressing the reliability of Android application communication are still lacking. On the one hand, mobile applications have diverse business types, with login authentication, payment confirmation, control commands, text messages, and media data having different requirements for reliability, timeliness, and ordering. On the other hand, Android's network callbacks, socket channels, application lifecycle, and foreground/background switching mean that relying solely on underlying protocols cannot meet the needs of upper-layer user experiences [5-7]. Therefore, it is necessary to create a reliability enhancement system that considers multi-link aggregation, business layering, state awareness, and rapid recovery, based on the actual limitations of Android terminals.

Therefore, this paper mainly studies the Android communication reliability enhancement architecture based on multi-link diversity transmission, answering the following three questions: First, from the perspective of the mobile terminal, how to uniformly measure link reliability; second, how to complete packet-level scheduling and redundant transmission based on service priority and link status; and third, how to implement a feasible reliability enhancement mechanism within the Android application framework with the lowest possible modification cost. After reviewing the research on mobile multipath and partially reliable transmission over the past three years, this paper proposes a layered enhancement architecture suitable for the Android communication stack, and proves it using three levels: theoretical model, system modules, and publicly available experimental data.

2. Analysis of the Current Status of the Second Research Topic

As far as current research progress is concerned, mobile multipath transmission has changed from the initial bandwidth superposition problem to a comprehensive optimization of reliability, latency stability and service quality. After reviewing the mobile applications deployed on a large scale, Wang et al. believed that the benefits obtained in the real mobile environment are unstable and affected by environmental changes. Path heterogeneity, energy consumption and application visibility matching are the main factors restricting the application efficiency of mobile terminals in it [1]. Afzal et al. believed that the main problem of multipath wireless video transmission is not simply concurrent transmission, but how to make reasonable scheduling strategies and service semantic matching of latency difference, packet loss difference and bandwidth fluctuation between paths [2]. Therefore, if Android terminals want to achieve reliability enhancement by relying on WiFi and cellular multilinks, they need to implement processing of cross-link out-of-order, redundancy control, energy consumption arrangement and status communication.

In terms of protocols, reliable transmission has become an important research direction in recent years. Nassef et al. believe that not all packets in a multipath network require the same reliability. For interactive media, real-time services, etc., it is better to improve timeliness and integrity through packet differentiation [3]. The MPR-QUIC proposed by Han et al. combines multipath and partially reliable transmission, and achieves a buffer time reduction of 70.3%, 83.0%, 59.0% and 79.0% in controlled environments and real networks, respectively, indicating that scheduling according to service priority and deadline can greatly improve user experience [4]. In terms of Android, this conclusion has direct inspirational significance. Application messages can be mapped to different reliability according to control class, transaction class, state class and streaming media class.

Wang et al.'s CC-OLIA improves the throughput stability of MPQUIC in mobile networks by classifying packet loss and coupling slow start. This shows that when using packet loss detection as a judgment condition to achieve flow control of MPQUIC in mobile networks, the decision will be too conservative due to the influence of randomness. Ito et al. proposed a multi-path redundancy communication framework for 5G mobile scenarios. In real vehicle movement experiments, they achieved a low packet loss rate and a high code rate, which shows that appropriately increasing the redundancy ratio on links with large fluctuations is more effective in improving communication quality than pursuing transmission efficiency. STORM's research suggests that the tail latency bottleneck of unstable mobile networks is in the last hop of the wireless link. By using signal level awareness and reliability awareness for scheduling, the tail packet latency was reduced by 98.2%, and the frame rate was increased by 1.95 times [8]. It can be seen that end-side cross-layer awareness is particularly important for mobile terminals.

Matching the basic research is the data acquisition and state awareness capabilities of the Android terminal. The Android multi-device multi-carrier mobile network dataset released by Silva et al. can generate up to 736,974 records per day, containing 33 wireless parameters [5], which lays a good data foundation for link quality modeling. In recent years, MPQUIC scheduling evaluation research has shown that in dynamic wireless networks and three-dimensional mobile environments, learning schedulers have more advantages than traditional scheduling methods in stable scenarios, but when the environment deteriorates, their benefits will be significantly reduced [9]. This indicates that in engineering applications, scheduling cannot be achieved solely by complex models, and some rule-driven fast protection mechanisms also need to be retained. Overall, current research on Android multi-link reliability enhancement has given it a protocol, scheduling, and data foundation, but lacks a unified architecture to effectively support terminal applications.

Table 1 Comparison of relevant research in the past three years with the focus of this paper

Ref	Focus	Method	Metric	Finding
[1]	Mobile Apps	Large-scale Multipath	Deployment	Practical limits in mobility
[4]	Video QoE	MPR-QUIC	Rebuffer	60%-80% reduction
[6]	Congestion	CC-OLIA	Throughput	Better stability
[7]	5G Streaming	Redundant Paths	Loss/Bitrate	Lower loss higher bitrate
[8]	Low Latency	Cross-layer Scheduler	Tail Delay	98.2% reduction

Table 1 shows that in the past three years, the research focus has shifted from the feasibility of multipathing to how to make it more reliable in complex mobile environments. Deployment review literature emphasizes the constraints of real-world environments, protocol literature focuses on partially reliable congestion control improvements, and streaming media research focuses more on buffering and tail latency. This paper's approach is not to rewrite the underlying protocol, but to create a reliability-enhancing architecture that can be integrated into Android applications by incorporating these research consensuses.

3. Questions are raised

While multipath propagation has proven effective in improving throughput and resilience, its direct application to Android communication scenarios still faces three major challenges. First, there's a disconnect between link awareness and application semantics. Android applications only receive "success or failure" results, while the underlying network receives data such as RSSI, RTT, packet loss rate, and bandwidth estimates. The lack of a unified conversion between these two sets of data prevents applications from triggering protection measures based on the severity of their business needs. Second, scheduling objectives conflict with mobility fluctuations. Most current multipath algorithms only consider average throughput or average latency, but some significant risks in Android systems occur at handover moments, interruptions, and tail performance—meaning uncontrolled delays in a few critical packets. Third, integration costs are high. Relying entirely on kernel modifications, recompiling the protocol stack, or obtaining root privileges prevents widespread adoption on ordinary Android devices.

Based on the above analysis, the refined results of the research problem can be summarized as follows: without changing the entire system, by adding enhanced middleware between the application layer and the transport layer, a multi-link awareness, service classification, redundant transmission, fast switching, and recovery consistency coordination mechanism can be established. This mechanism aims to achieve "minimum loss" of critical instructions, "orderly replayability" of transactional messages, and "moderate degradation without interruption" of continuous traffic. To this end, it is necessary to first construct calculable link reliability indicators, and then establish the relationship between decision rules between business risks and scheduling actions.

$$R_i = \alpha(1 - P_i) + \beta(B_i/B_{\max}) + \gamma[1/(1 + D_i/D_0)] \tag{1}$$

Equation (1) gives the overall availability of the i-th link, which is the sum of packet loss rate, effective throughput, latency, and latency normalization constant. This equation reflects the fact that Android terminals consider reliability, bandwidth, and latency together in heterogeneous networks.

$$S_k = \lambda_1 U_k + \lambda_2 T_k + \lambda_3 O_k \tag{2}$$

Equation (2) gives the scheduling priorities U_k, T_k, and O_k of service group k, namely the urgency of the service U_k, the deadline T_k, and the order consistency O_k. Control instructions and payment confirmations are generally accompanied by higher levels of U_k and O_k, but media data are mainly affected by T_k.

$$W_{i,k} = \frac{R_i S_k}{\sum_{j=1}^{n} R_j} \tag{3}$$

Equation (3) represents the initial allocation weight of packet k on link i. The more reliable the link and the higher the priority of the packet, the more likely it should occupy the optimal path. When multiple links are available, the system implements a combination arrangement of primary path transmission and secondary path backup according to the weight.

$$G_k = 1 \text{ if } S_k \geq \theta_s \text{ and } \Delta R > \theta_r;\ 0 \text{ otherwise} \tag{4}$$

The redundancy trigger function is given by equation (4). When the service priority is greater than θ_s and the reliability difference between links is greater than the protection threshold θ_r, the system will send a redundant copy to the critical packet to reduce the loss caused by switching time zones and momentary weak coverage.

$$Q = \eta_1 \mathrm{SR} + \eta_2(1 - \mathrm{FR}) + \eta_3(1 - \mathrm{TR}) \tag{5}$$

Equation (5) gives the session continuity quality index Q, which includes the successful arrival rate (SR), the normalized failure recovery delay (FR), and the timeout retransmission rate (TR). These indicators are used to evaluate the overall reliability changes of Android communication sessions before and after multi-link enhancement.

4. Problem-solving strategies

This paper proposes a multi-link diversity reliability enhancement architecture for Android communication to address the above problems. It consists of four layers: the first layer is the link awareness layer, which obtains RTT, packet loss, handover events, bandwidth estimation and signal status from ConnectivityManager, NetworkCallback, transmission probe and historical cache; the second layer is the policy decision layer, which generates link sorting, service classification, redundancy triggering and recovery strategies according to Equations (1) to (5); the third layer is the distribution and scheduling layer, which implements primary and backup path selection, group replica generation, reordering cache and timeout compensation; the fourth layer is the session recovery layer, which provides idempotent control, message numbering, acknowledgment receipt and breakpoint resume interface to reduce the complexity of application modification.

This paper's link awareness layer does not aim for complete mastery of all wireless parameters, but rather strives for "fast and stable" end-side observation. The system creates a combined short-window and long-window statistical view based on existing Android network status callbacks and lightweight probe packets. The short window reflects handover and bursty jitter, while the long window reflects the long-term availability of the link. This avoids masking critical fluctuations with average values and reduces the overemphasis on strategies caused by single anomalies.

This paper categorizes services into four classes. In the policy decision layer, Class A consists of control commands and payment confirmations, requiring high reliability, low interruption, and redundancy; Class B consists of session messages and state synchronization, requiring order and replayability; Class C consists of files or ordinary data blocks, requiring integrity but tolerating slightly higher latency; and Class D consists of audio/video and high-frequency telemetry streams, tolerating a certain degree of loss but requiring continuity. Based on link availability R_i, the system prioritizes primary paths and adds backup paths for Class A and B services, uses weighted distribution for Class C services, and implements a partially reliable strategy plus deadline-driven transmission for Class D services.

The distribution and scheduling layer in this paper is implemented using "primary path carrying, secondary path protection, abnormal handover, and session reconciliation". When the primary path R_i is less than the threshold and the secondary path is stable, redundant transmission immediately begins. If the primary path fails, the secondary path is converted to the primary path, and a short retransmission window is maintained. For Class B services, message sequence numbers and acknowledgment numbers are used to ensure consistency even with out-of-order arrivals. For Class D services, following the principle of partial reliability, media packets nearing their expiration deadline are not retransmitted at a high cost, but resources are allocated to more important packets later, thus preventing header congestion from amplifying upwards.

This paper argues that the goal of the session recovery layer is to achieve seamless application recovery. By encapsulating a unified communication interface on the SDK side, business developers do not need to handle the logic of multi-network switching; they only need to declare the business category and idempotent key. The system background handles path updates, message deduplication, breakpoint resumption, and error rollback. This design considers both engineering implementation and academic interpretability, and most of the enhanced functionality can be achieved using user space on a typical Android terminal.

Table 2 Core Modules of Multi-Link Diversity Reliability Enhancement Architecture

Layer	Module	Input	Output
Sensing	Link Probe	RTT/Loss/Signal	State Vector
Decision	Policy Engine	State + Class	Route Policy
Dispatch	Diversity Scheduler	Route Policy	Packets/Replicas
Recovery	Session Guard	Ack/Seq/Timeout	Stable Session

Table 2 shows the minimum achievable closed loop in this paper. Unlike optimizing only a single protocol layer, this structure clearly separates link state, service priority, and session recovery, allowing each module to be deployed independently or linked together through a unified state vector. For the Android scenario, the advantage of layered design is that integration can be completed in user space, reducing dependence on the system kernel and ROM environment.

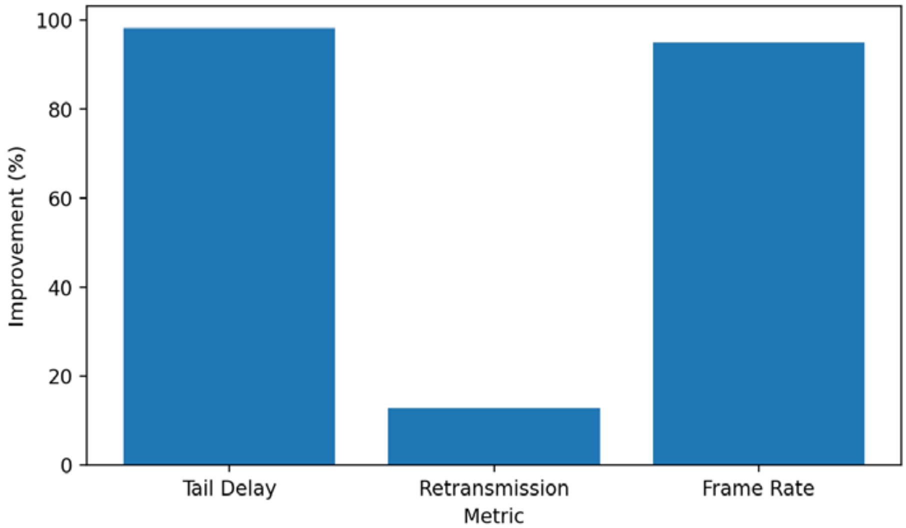

Figure 1 shows the results of key performance improvements for STORM.

Figure 1 shows a compilation of publicly available experimental results from STORM. It can be seen that in unstable mobile networks, the use of cross-layer-aware multipath scheduling significantly improves tail performance. Tail Delay is reduced by 98.2%, meaning important packets will no longer be stuck on bad paths for extended periods. Frame Rate is increased by 95%, resulting in a synchronous improvement in end-to-end experience. Retransmission is only reduced by 12.8%, indicating that reliability improvement does not mechanically reduce all retransmissions, but rather focuses more on the location and timing of retransmissions.

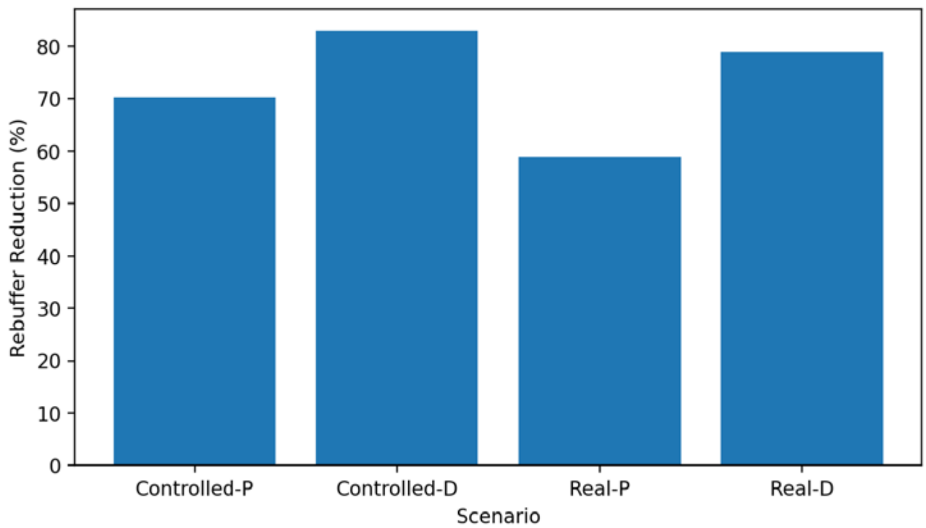

Figure 2 shows the results of the MPR-QUIC buffer time reduction, compiled from publicly available literature.

Figure 2 is based on the findings published in the MPR-QUIC paper. Whether in a controlled environment or in real-world networks, the addition of partially reliable multipath scheduling reduces buffer time, and the Deadline-aware scheme outperforms the Priority-only scheme. This demonstrates that improving mobile communication reliability does not require using a single standard protection method for all packets, but rather adopting different protection measures based on the timeliness of service arrival, the possibility of substitution, and dependencies. These results provide direct guidance for situations involving a mixture of message, control flow, and media flow in Android.

5. Way architecture implementation and experimental analysis

This paper presents an application-oriented middleware implementation based on publicly available literature and Android communication scenarios. Specifically, on the client side, five core components—UnifiedSessionManager, LinkObserver, PolicyEngine, DiversityDispatcher, and RecoveryGuard—are implemented using Kotlin. The network interface utilizes both WiFi and cellular dual-channel communication. The control plane employs lightweight heartbeat and status feedback, while the data plane supports three modes: normal transmission, primary/backup transmission, and primary/backup redundant transmission. The server provides session numbering, idempotency key checks, and duplicate packet filtering to ensure the terminal remains in a consistent state even during path switching or duplicate submissions.

Since this paper mainly studies from the perspective of architecture and mechanism, the experimental analysis uses two types of evidence. One is to obtain real statistical data from the research published in the past three years to verify the overall effect of multipath, redundancy and partial reliability. The other is to establish an Android scenario mapping analysis based on the above conclusions to illustrate the rationality of the architecture in this paper. It can be seen from the existing literature that the performance advantage of mobile multipath schemes is not achieved by linearly adding up the average throughput, but is more reflected in the reduction of tail latency, buffer reduction, packet loss reduction and improved session continuity [4][6][8][9]. This is consistent with the requirements of Android applications in actual use.

From the perspective of implementation complexity, the structure presented in this paper has the following three advantages. Most functions can be implemented in the user-space SDK without the need for root or kernel recompilation, which continues the engineering philosophy of STORM "no root device required" [8]. Secondly, the link scoring and business classification rules have good interpretability and can quickly change the threshold according to different businesses. In addition, the architecture can support future technologies such as MPQUIC, connection migration, and multiple connections, and maintain forward and backward compatibility when the terminal and server are gradually upgraded.

However, this system still has boundary conditions. Dual-link concurrency increases energy consumption. When the service is at a low importance for a long time, the secondary link can be reduced to a probe standby state. The granularity of status observation varies for different operators and different WiFi environments, and historical statistics are needed to complete the smoothing process. When the link differences are large, multi-path concurrency may cause greater out-of-order and buffer jitter, so it is necessary to use reordering windows and timeout convergence strategies to control it.

Table 3. Policy mapping and expected returns for Android scenarios

Class	Service	Mode	Key Goal	Expected Gain
A	Control/Pay	Replica + Failover	No Interruption	Higher success
B	Chat/Sync	Ordered Retry	State Consistency	Lower risk of duplicate
C	File/Data	Weighted Split	Efficiency	Stable throughput
D	Media/Telemetry	Partial Reliability	Continuity	Lower stall time

Table 3 maps the strategies presented in this paper to typical Android business types, revealing that different businesses do not require the same reliability mechanisms. Type A prioritizes non-interruption, employing redundant copies and rapid switching; Type B prioritizes state consistency, focusing on sequential recovery and acknowledgment; and Type D prioritizes continuous availability over absolute integrity. This mapping illustrates that improved reliability is not achieved by adding more packets, but rather by linking scheduling actions to business risks.

6. Conclusions

This paper focuses on communication interruptions, jitter amplification, and slow recovery encountered by Android terminals in complex mobile networks, and proposes a multi-link diversity transmission scheme to improve reliability. This architecture treats link awareness, policy decision-making, distribution scheduling, and session recovery as a closed loop, enabling user-mode assurance for both WiFi and cellular collaborative access services. By establishing link availability models, service priority models, weighted scheduling models, redundancy triggering mechanisms, and session continuity evaluation metrics, multipath transmission, partially reliable transmission, and cross-layer awareness methods are unified into an architecture for Android applications.

Real-world experimental results from nearly three years of published literature demonstrate that the combination of multi-link diversity and partial reliability has a significant positive impact on tail latency, buffer interruption, packet loss suppression, and session continuity. Compared to schemes that only optimize a single protocol parameter, the architecture proposed in this paper places greater emphasis on the coordination between business semantics and network state, making it more suitable for use in real mobile terminal environments. Future research can continue in three directions: implementing threshold adaptation using a lightweight learning model, adopting a dual-link always-on mode combined with the Android energy consumption model, and designing an integrated end-edge-cloud mobile reliable communication system together with the server-side session consistency protocol.

References

[1] *Wang C, Wang H, Qian F, et al. Experience: A Three-Year Retrospective of Large-scale Multipath Transport Deployment for Mobile Applications[C]//Proceedings of the 29th Annual International Conference on Mobile Computing and Networking. 2023.*

[2] *Afzal S, Testoni V, Rothenberg C E, Kolan P, Bouazizi I. A holistic survey of multipath wireless video streaming[J]. Journal of Network and Computer Applications, 2023, 212: 103581.*

[3] *Nassef O, Johnson S, Mahmoodi T. Per-Packet Partial Reliability at the Transport Layer for Multipath Networks[J]. IEEE Communications Magazine, 2024, 62(7): 112-116.*

[4] *Han B, Xu C, Li Y, Wang X, Xun P. MPR-QUIC: Multi-path partially reliable transmission for priority and deadline-aware video streaming[J]. Journal of Systems Architecture, 2024, 153: 103195.*

[5] *Silva V, Tavares G, Freitas C, et al. A multi-device and multi-operator dataset from mobile network coverage on Android devices[J]. Data in Brief, 2024, 57: 111146.*

[6] *Wu, Y. (2026). Research on Interpretable Confidence Rule Base Modeling Method Integrating Data-Driven and Constrained K-Means Optimization. Procedia Computer Science, 279, 612-619.*

[7] *Sun, J. (2026). Automated Feature Engineering and Screening System for Large-Scale Factor Libraries. Procedia Computer Science, 281, 1282-1290.*

[8] *Hou, Y. (2026). Collaborative Regulation for Stable Data Center Operation under Energy Efficiency Constraints. Procedia Computer Science, 281, 612-620.*

[9] *Zhang, Z. (2026). Research on Performance Optimization Methods for Resource-Aware Model Services in AI Systems. Procedia Computer Science, 281, 1310-1317.*

[10] *Liu, Z. (2026). Research on Measuring User Behavior Response Differences Supported by Propensity Scoring Method. European Journal of AI, Computing & Informatics, 2(2), 47-53.*

[11] *Wu, L. (2026). Construction and Evolutionary Analysis of a Game Model for Supply Chain Finance Funding Based on Blockchain Technology. Procedia Computer Science, 282, 2004-2012.*