

The background features abstract blue geometric shapes, including triangles and polygons, in various shades of blue, creating a modern and dynamic look.

# ゲームリアルタイム通信プロトコル 第四回会合

# 「ゲームのリアルタイム通信に必要な要素の整理」の続き

# リアルタイム通信まわりの現状整理

- ▶ プロトコルの整理 ← 第三回で実施
- ▶ 実装の整理
  - ▶ ライブラリ(OSS)
  - ▶ ゲームエンジン
  - ▶ 通信エンジン
  - ▶ プラットフォーム

# プロトコルの整理：宿題①

「疎通性」の呼び方について

## ▶ (前回より)

### ▶ P2P の接続性と分けて議論するのは良い話

▶ もう少し分かり易い用語の定義ができると良さそう

▶ 「疎通性」という言葉だとイメージがぱっと浮かび辛い

## ▶ 「(プロトコルの)到達性」はどうでしょうか？

### プロトコル：疎通性

- ▶ ミドルボックスにドロップされない等プロトコルそのものの疎通性
  - ▶ P2P における接続性とは別
- ▶ WAF やロードバランサ等に広く対応している
  - ▶ 別の指標として扱った方が良いかも？
- ▶ フォールバックの話もある
  - ▶ QUIC on Stream

# プロトコルの整理：宿題②

- ▶ 「必要な要素」について「仕様」、「実装」、「環境」の分類別に分けてまとめた方が良さそう
  - ▶ 実装例：コンシューマプラットフォームに移植し易いか
  - ▶ 環境例：疎通性の高低
- ▶ 今回の議論をベースに次回までに主幹がまとめておく

# プロトコルの整理：宿題②

作成中……

仕様

実装

環境

- 到達性(疎通性)
- 暗号化
- 制御アルゴリズム
- 到達保証
- リオーダー
- 優先度
- CM (Connection-Migration)
- Multipath

- 暗号化
- 難読化
- 移植容易性
- 実装難易度
- 言語

- 到達性(疎通性)
- 中継(Pub-Sub)

# 現状整理：実装

- ▶ ライブラリ (OSS)
- ▶ ゲームエンジン
- ▶ 通信エンジン
- ▶ プラットフォーム



# ライブラリ(OSS)

## ▶ Enet

- ▶ RUDP (C言語)
- ▶ ゲームに特化した設計
- ▶ どちらかと言えば軽量で中規模タイトルまで
- ▶ 代表的な採用タイトル: たくさん

## ▶ NetCode IO

- ▶ RUDP (C言語)
- ▶ ゲームに特化した設計
- ▶ 専用サーバを用意する等セキュリティを意識(大規模タイトル向け)
- ▶ 代表的な採用タイトル: たくさん

## ▶ RakNet

- ▶ RUDP (C/C++)
- ▶ 優先度制御もある
- ▶ 本家では開発は止まっているがForkや派生開発が継続(別言語版も)
- ▶ 代表的な採用タイトル: Minecraft 統合版



# ライブラリ (OSS)

## ▶ KCP

- ▶ 様々な言語への移植版がある
- ▶ 信頼性のある ARQ(Automatic Repeat request) プロトコル
  - ▶ TCP とは異なる自動再送ポリシーを採用
  - ▶ モバイルネットワークに特化してチューニングしてある
- ▶ Pure C# 実装もある
  - ▶ <https://github.com/MirrorNetworking/kcp2k>
  - ▶ <https://mirror-networking.gitbook.io/docs/manual/transport/kcp-transport>
- ▶ 代表的な採用タイトル：原神 (他中国のタイトルで多く採用)

# ライブラリ (OSS)

## ▶ TQUIC

- ▶ Tencent が開発中の QUIC 実装 (Rust 製)
- ▶ マルチプラットフォーム対応予定
  - ▶ 現状は Linux、MacOS、FreeBSD、Windows、iOS、Android、Harmony
  - ▶ 「TQUIC は、Rust がコンパイルできるほぼすべての環境で動作します。Rust/C/C++ 用の API を提供します。」
- ▶ パフォーマンスに寄せている
  - ▶ [https://tquic.net/docs/further\\_readings/benchmark/](https://tquic.net/docs/further_readings/benchmark/) に計測結果がある
    - ▶ 計測条件等が不明瞭なので現状ではそこまで信ぴょう性は高くない
  - ▶ blogには今後論文発表予定とあるがまだ動き無し
    - ▶ <https://tquic.net/blog/tquic-open-source>
    - ▶ blog は 2023/10 以降更新無し (リポの開発は活発)
- ▶ QUIC, HTTP/3 のみに実装(フォールバック無し)

# ライブラリ (OSS)

## ▶ [MagicOnion](#)

- ▶ 軽量な RPC フレームワーク (C#)
- ▶ 代表的な採用タイトル: たくさん

## ▶ [YetAnotherHttpHandler \(Cysharp\)](#)

- ▶ grpc-dotnet を Unity で使う為の HTTP/2 ハンドラ (C#)
- ▶ MagicOnion の HTTP ハンドラとして使うことも可能
- ▶ 代表的な採用タイトル: メルクストーリア

## ▶ [KcpTransport \(Cysharp\)](#)

- ▶ KCPのPure C#実装
- ▶ 現在  $\alpha$  版

# ライブラリ (OSS)

- ▶ [LiteNetLib](#)

- ▶ RUDP (C#)
- ▶ Windows, Linux, Android, iOS 以外に Switch にも対応(とある)
- ▶ 代表的な採用タイトル: 情報不足なのでご存じの方居れば

- ▶ [WSNet2 \(KLab\)](#)

- ▶ WebSocket ベース (C#)

# ゲームエンジン

## ▶ Unity

- ▶ (g)RPC, QUIC, HTTP/3のような最新のプロトコルに標準で対応していない
- ▶ Unity Transport Packageというレイヤーはある
  - ▶ UDPベースでRUDP相当の機能
- ▶ CoreCLR 対応により .NET の QUIC や HTTP/3 を利用可能になる可能性がある
  - ▶ .NET の QUIC は MsQuic ベースなのでマルチプラットフォーム対応がどうなるかは未知数

# ゲームエンジン

※前回の共有から特に変化無し

## ▶ Unreal Engine

- ▶ RPCはあるが生のRUDPはない
  - ▶ RPC上でunreliableな通信は可能
- ▶ QUICに対応中
- ▶ HTTP/3に非対応
- ▶ EOSはUEに紐づかないでも使えるので通信エンジン側に記載

## ゲームエンジン : Unreal Engine QUIC 補足

※前回の共有から特に変化無し

- ▶ MsQuic を使用した実装が用意されている
  - ▶ Engine¥Source¥ThirdParty¥MsQuic に配置されている
  - ▶ いつのバージョンからかは不明
- ▶ MsQuic.Build.cs を見る限り Windows/Linux/Mac までの対応
- ▶ ヘッダを見た感じとりあえず突っ込んでラッパした程度に見える

# ライブラリ(ゲームエンジン向けOSS)

## ▶ [Mirror](#)

- ▶ Unity 向けのマルチプレイヤー対応通信ライブラリ
- ▶ UNET をベースに作成されている
- ▶ RUDP まで
  - ▶ 実装は前述した KCP-transport
- ▶ 派生で [MirageNet](#) というものもある

## ▶ [ue-quic](#)

- ▶ UE 向け CloudFlare quiche のラッパ
- ▶ コンシューマは非対応



## 通信エンジン

- ▶ Diarkis
  - ▶ RUDP まで
  - ▶ マルチプラットフォーム対応
  - ▶ 採用タイトル: TEKKEN 8/プロジェクトセカイ カラフルステージ! feat. 初音ミク
    - ▶ 未公開の採用タイトルも多数

## 通信エンジン

### ▶ Photon

- ▶ RUDP, WebSocket あたりに対応 (Photon RealTime)
- ▶ コンシューマも含むマルチプラットフォーム対応
- ▶ 用途に応じたラインナップを用意
  - ▶ オブジェクト同期等の機能も揃っている
- ▶ サーバの無料ライセンスはなくなってしまった
- ▶ 主な採用タイトル : VR Chat, PICO PARK 2
  - ▶ 国内タイトルでの採用も多くあるはずだが事例が見つからない

# 通信エンジン

## ▶ STRIX ENGINE

- ▶ RUDP まで
- ▶ マルチプラットフォーム対応
- ▶ 採用タイトル：ブループロトコル

## ▶ モノビット

- ▶ MRU : Monobit Reliable UDP 採用？
  - ▶ 参考 : MRU : [Monobit Reliable UDP ~5G世代のモバイルゲームに最適な通信プロトコルを目指して~](#)
- ▶ コンシューマタイトル実績無し？ (PS Vitaは確認)
- ▶ 主な採用タイトル：リボルバーズエイト

# 通信エンジン

- ▶ Epic Online Services
  - ▶ P2P の通信の標準プロトコルは DLTS
  - ▶ 任意のプロトコルに差し替えも可能
    - ▶ [地球防衛軍6の事例](#)
  - ▶ (プロトコルから外れるが) サーバ費用を Epic 側が持ってくれるのも大きな特徴
  - ▶ サーバ費用を自前で持てない小規模のタイトルは通信エンジン(サービス)として EOS を選択することが多い

# プラットフォーム

## ▶ MsQuic (MicroSoft)

- ▶ GDK(Windows, Xbox)で利用可能

- ▶ [https://learn.microsoft.com/ja-jp/gaming/gdk/\\_content/gc/networking/overviews/game-mesh/msquic-intro-networking](https://learn.microsoft.com/ja-jp/gaming/gdk/_content/gc/networking/overviews/game-mesh/msquic-intro-networking)

- ▶ Schannel 部さえ外せば Windows の依存は外せる

- ▶ <https://x.com/gamernb/status/1611141009488089090>

## ▶ Android, iPhone

- ▶ ゲーム向けではないが QUIC や HTTP/3 は標準で実装されている
- ▶ 言語的にマルチプラットフォームの共通実装に用いるのは厳しいので、別途 C/C++ で実装したものをを用いる戦略が一般的

# プラットフォーム

## ▶ GameNetworkingSockets (Valve)

- ▶ 独自の RUDP
  - ▶ 非ストリーム
  - ▶ QUIC や DCCP を元にした再送制御
  - ▶ 信頼性の有無の柔軟な切り替え
- ▶ IsPS5 や IsX360 等の定義がある
  - ▶ コンシューマ対応ソースの配布は別途問い合わせらしい
- ▶ 国内だと Cygames 様の Cyllista Game Engine に採用
  - ▶ <https://magazine.cygames.co.jp/archives/27482>

会合での情報交換以外に当会合で  
できることがないか

# 過去の議論からの抜粋

- ▶ リアルタイム通信ゲームサンプルの作成
- ▶ QoE 計測のオンラインゲームのテストモデル
  - ▶ 特定のプロトコルを用いたオンラインゲームのテストモデルのようなものをゲーム業界から提供して、「対戦中の同期待ちフレーム数」とか「画面遷移でのウェイト発生回数」というような QoE 計測モデルを提示したい
- ▶ TWAMP のゲーム向けの実装に向けた活動
- ▶ 暗号化のユースケースまとめ
  - ▶ プロトコルの整理側で全体のユースケースをまとめる方向の方が筋は良さそう



# 主幹からの連絡

## ▶ 次回開催

- ▶ IETF 122 (3/15-21) 終了翌月-翌々月
  - ▶ 2025/4 - 2025/5
    - ▶ CEDEC 2025 の講演情報が出揃ってから理想
- ▶ 会場をお貸し頂ける会社様を引き続き募集！

# スケジュール(再掲)

2023年

当会号の発足 (CEDEC 2023)

2024年

①ゲーム業界のリアルタイム通信プロトコルにおける課題を洗い出す ← イマココ

2025年

②QUIC等標準のプロトコルを用いた場合の検討・議論・事例収集 ← ここが少しずれそう……

2026年

③ゲームのリアルタイム通信プロトコルの標準化を進める

2027年

2028年

※実際には①-③の要素はある程度平行で進む想定