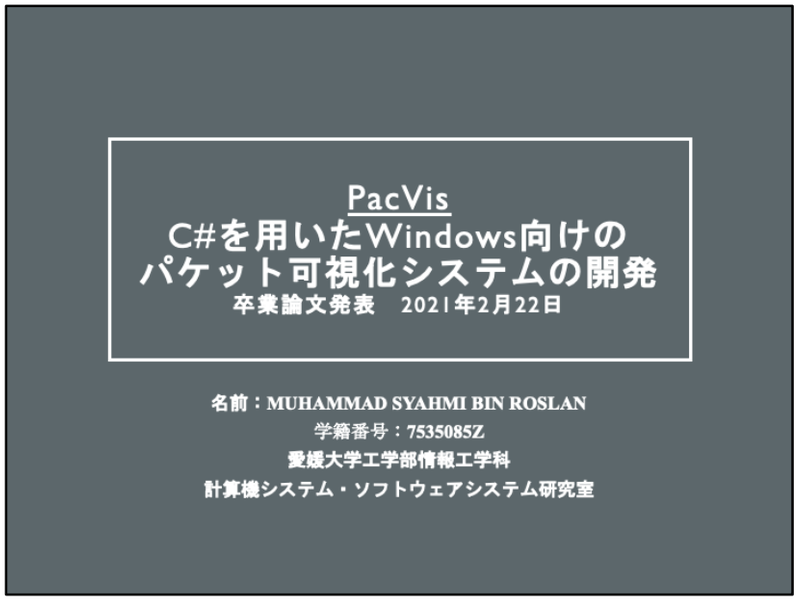
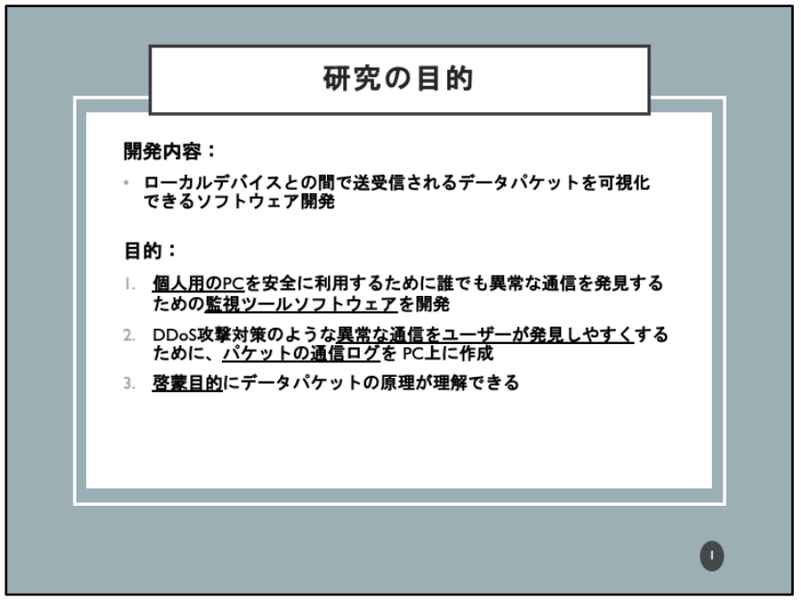


C#を用いたWindows向けのパケット可視化システムの開発に関する題しましてムハマドが発表させて頂きます。

どうぞよろしくお願いいたします。

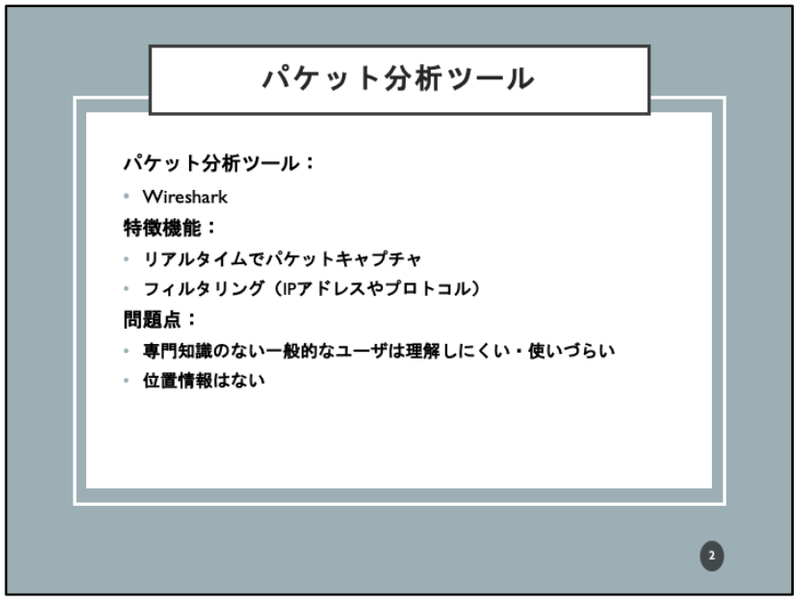


本研究内容としては、ローカルデバイスとの間で送受信されるデータパケットを可視化できるソフトウェアを開発します。

この研究の目的として、
まず、個人のPCを安全に利用するために誰でも異常な通信を発見するための監視ツールソフトウェアを開発し、

DDoS攻撃対策のような異常な通信をユーザーが発見しやすくするために、パケットの通信ログを PC上に作成するソフトウェアを開発する目的があります。

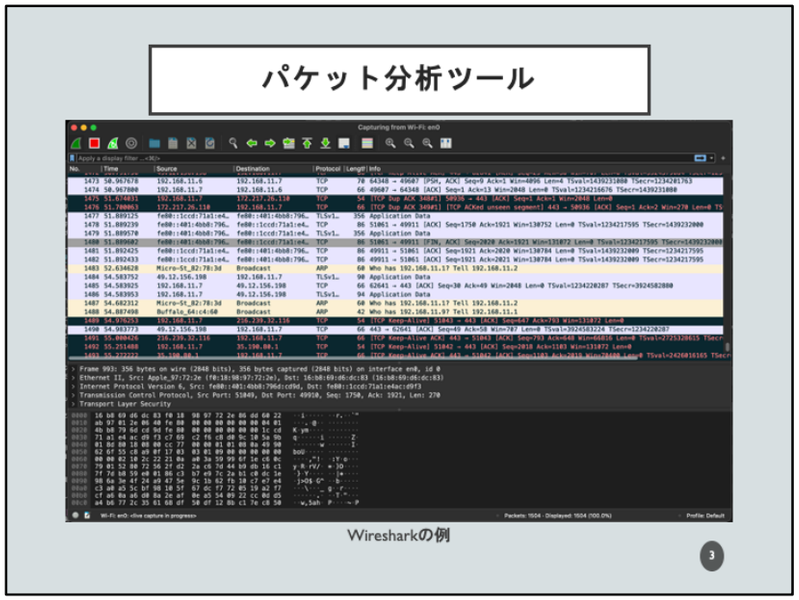
最後には、啓蒙（けいもう）目的にデータパケットの原理が理解できる、学べるソフトウェアを開発します。



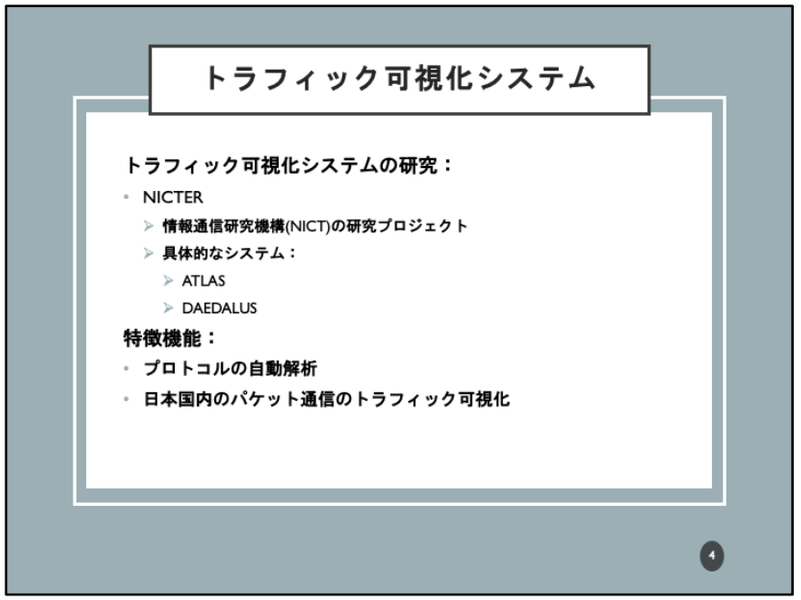
パケットの送受信状況を見る手段としてはWiresharkのようなパケット分析ツールがあります。

Wiresharkはリアルタイムでパケットキャプチャ機能があり、それに対するフィルタリング機能も対応します。

ただし、Wiresharkは、一般的なユーザは理解しにくいし、使いづらいシステムであり、位置情報も対応しないという問題点があります。



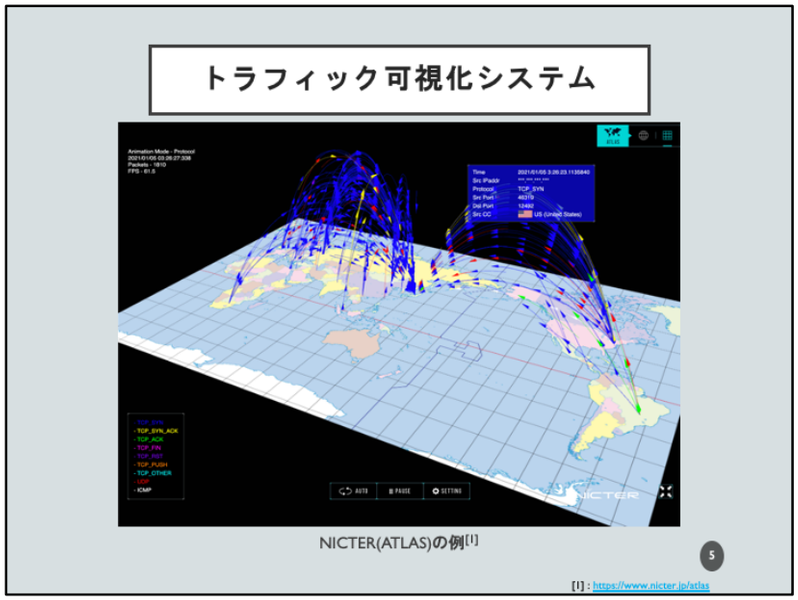
これは起動中のWiresharkの例です。ご覧になっている通り、Wiresharkは専門家向けのシステムであります。



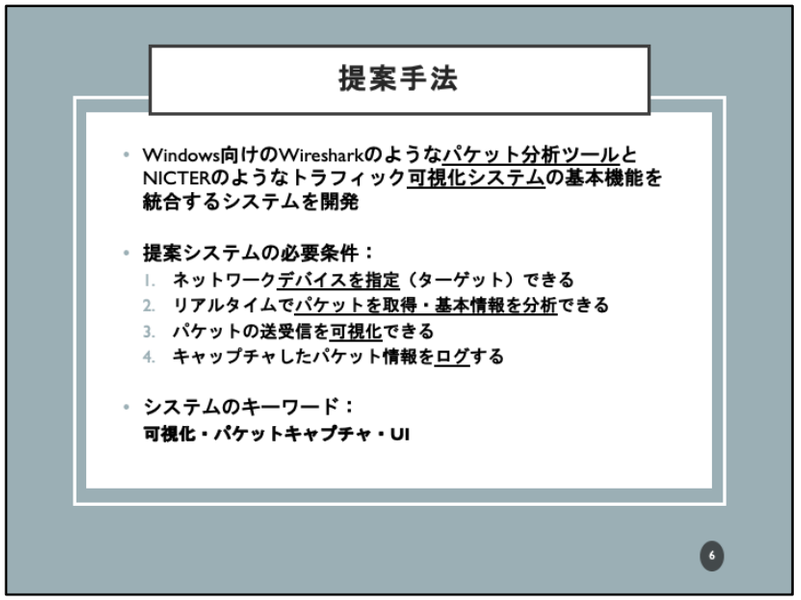
日本国内だとNICTERというトラフィック可視化システム研究プロジェクトがあります。

NICTERに関係する具体的なシステムはATLASとDAEDALUSがあります。

これらのシステムは、プロトコルの自動解析しながら、日本国内のパケット通信のトラフィック可視化できる特徴があります。



例としてNICTERのATLASを示(しめ）します。

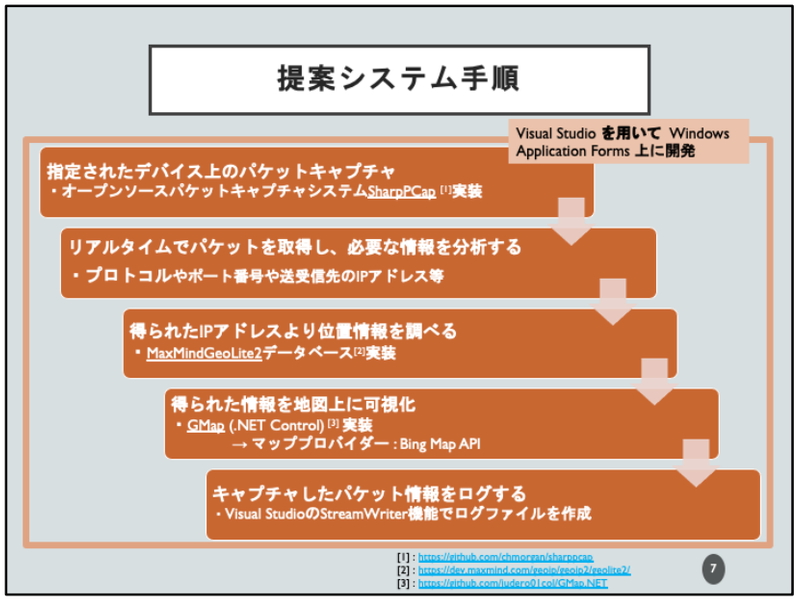


本研究の提案手法に移りますと、Windows向けのWiresharkのようなパケット分析ツールとNICTERのようなトラフィック可視化システムの基本機能を統合するシステムを開発します。

このシステムを開発するために、必要条件は4つあります。

それは、ネットワークデバイスを指定（ターゲット）できること、パケットを取得・情報を分析できること、送受信を可視化できることと、得られた情報をログすることです。

// 本システムのキーワードとしては可視化・パケットキャプチャとUIとなります。



提案ソフトウェア手順はこのようになります。

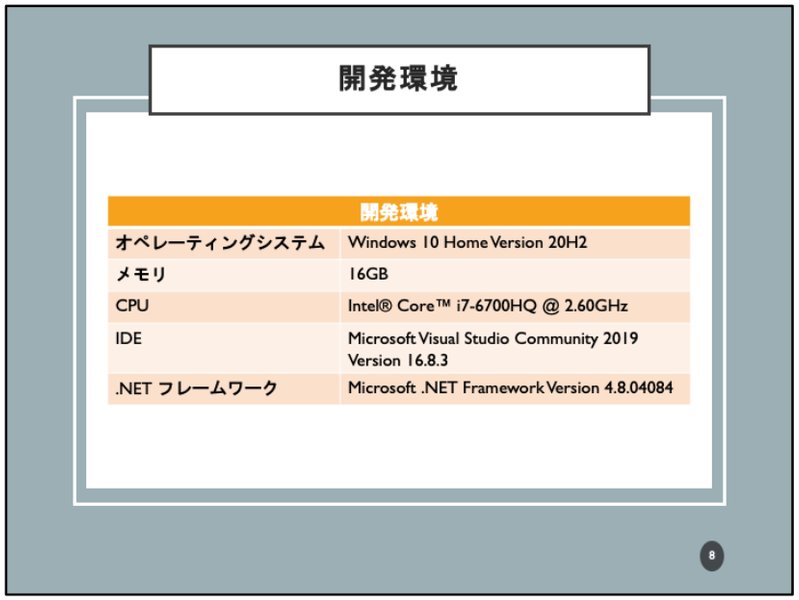
まず、オープンソースパケットキャプチャシステムのSharpPCapを利用して、指定されたデバイス上のパケットをキャプチャし、リアルタイムでパケットから必要な情報を分析します。

そして、MaxMindGeoLite2データベースでその位置情報を調べます。

得られた位置情報をGmapの.NET Controlを実装し、地図上に可視化をします。

最後に、Visual StudioのStreamWriter機能でキャプチャしたパケット情報をログします。

この流れではシステムの基本アルゴリズムと考えられ、Visual Studioを用いてWindows Application Forms 上に開発します。

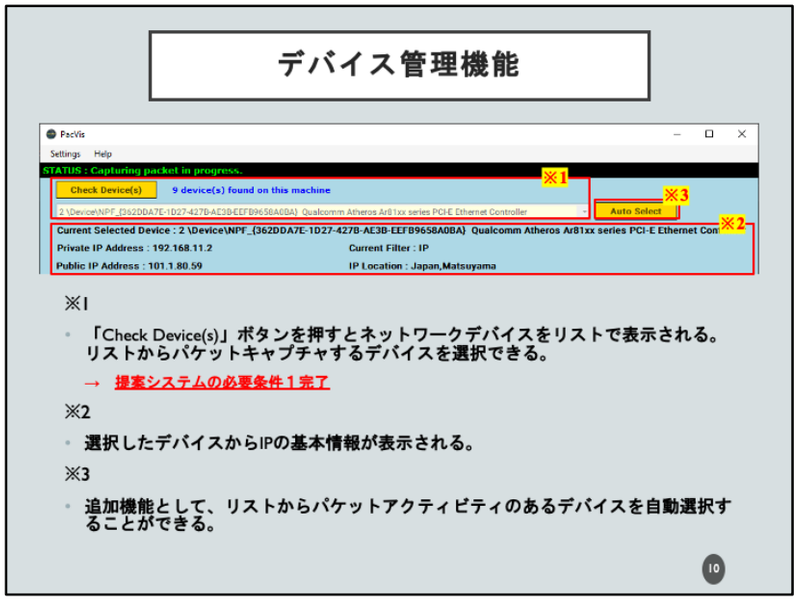


開発環境はこのようになっています。

結果としては、このようなシステムが開発しました。

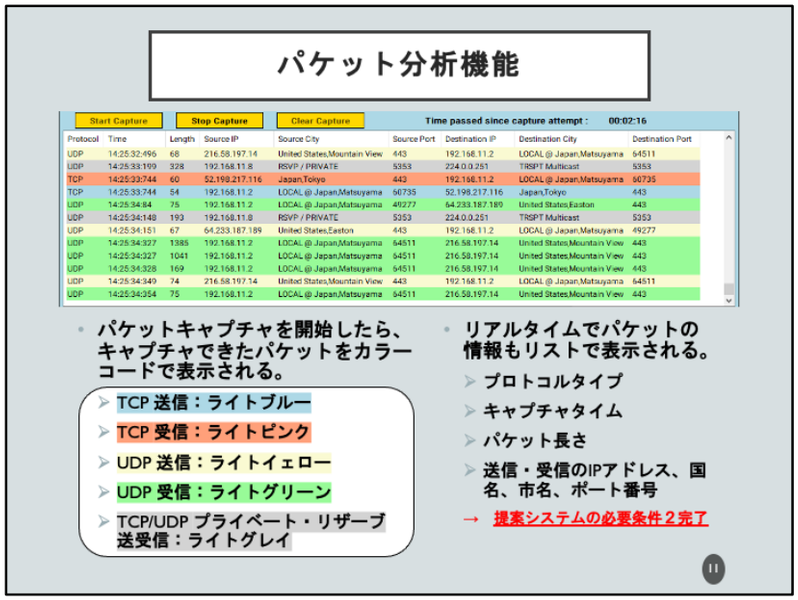
なお、本研究ではこのシステムはPacVisという名前がつけられます。

PacVisを説明すると、大きくの3つの部分に分けることができます。



まず、デバイスを管理する部分です。

ポイント１を中心となり、「Check Device(s)」ボタンを押すとネットワークデバイスをリストで表示され、デバイスを選択できる部分は提案システム必要条件１が達成しました。



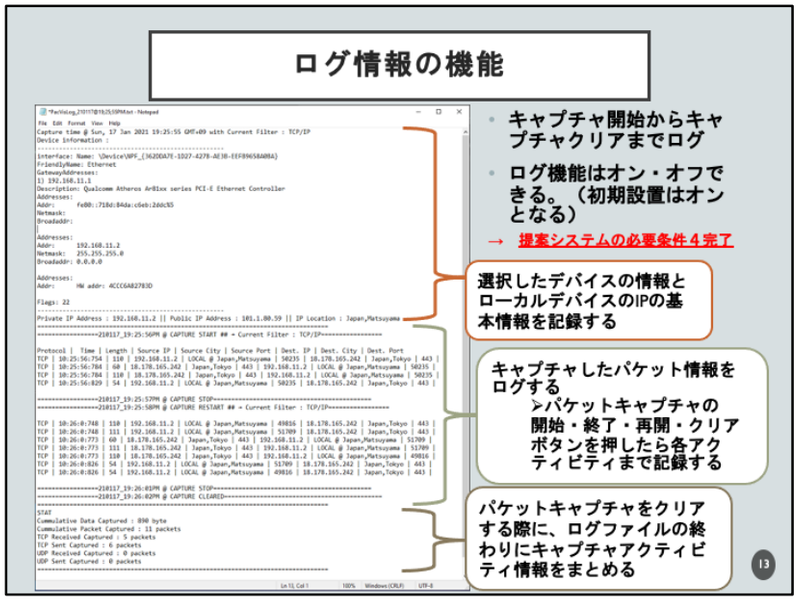
次、パケット分析の部分です。

ここでは、リアルタイムでパケットの情報をカラーコード付きリスト型で表示され、提案システム必要条件２を完成しました。

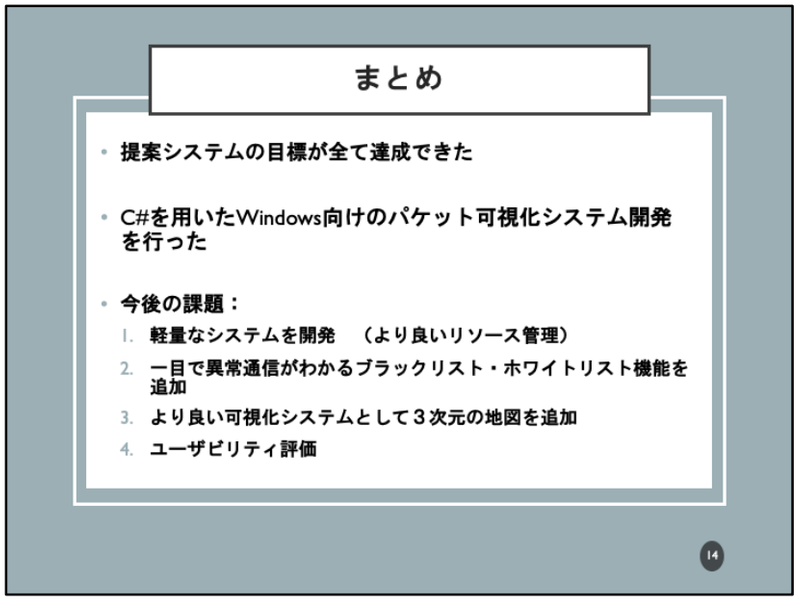
最後に、PacVisの可視化の部分があります。

ご覧になる通り、ポイント４はパケット通信に従って「最新件」順にTCPパケットの送受信をリアルタイムで可視化できます。

これで提案システム必要条件３が達成しました。



キャプチャ開始からクリアまで全てのアクティビティがログファイルを自動的に作成し、提案システムの必要条件４が完成できたと考えられます。



まとめて、提案システムの目標が全て達成でき、C#を用いたWindows向けのパケット可視化システム開発が行いました。

今後の課題としては、より良いリソース管理ができる軽量なシステムに改善（かいゼン）できるかという疑問点があります。

そして、一目で異常通信がわかるようにブラックリスト・ホワイトリスト機能を、より良い可視化システムとしてz３次元の地図を追加できれば、UI的に向上できると思います。

今回は新型コロナウイルスでユーザビリティテストができなかったため、ユーザビリティテストも今後の課題にしました。

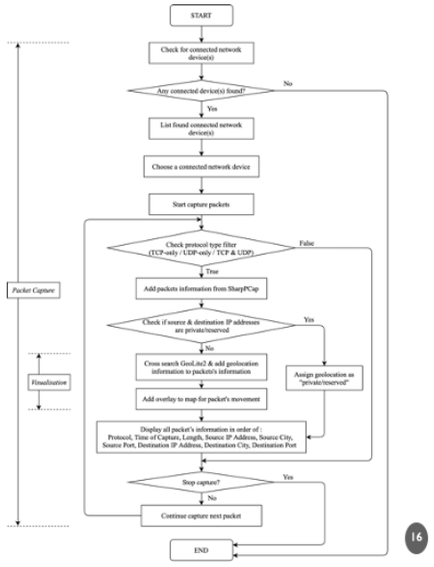
以上で発表終わります。
ご静聴ありがとうございました。

補足：UDPの可視化問題

- UDPはTCPと違って、プライベート・リザーブIPアドレスが非常に多い。
- 可視化にすると、ゲオロケーション情報を必要だが、プライベート・リザーブIPアドレスはデータベースを調べてもゲオロケーション情報を取得できない。
- ゲオロケーションが取得できなければ、システムはエラーと表示されてしまう。
- この問題より、UDPを可視化にすると、例外条件を多数追加必要で、安定なシステムを開発できたと判断するため、スタビリティテストをしなければならない。（時間の問題）
- 加えて、角例外を調べると、アルゴリズムの複雑度が高めてしまって、処理時間が増える可能性がある。

15

補足：パケット
キャプチャの
フローチャート



補足：デバイス
自動選択機能の
フローチャート