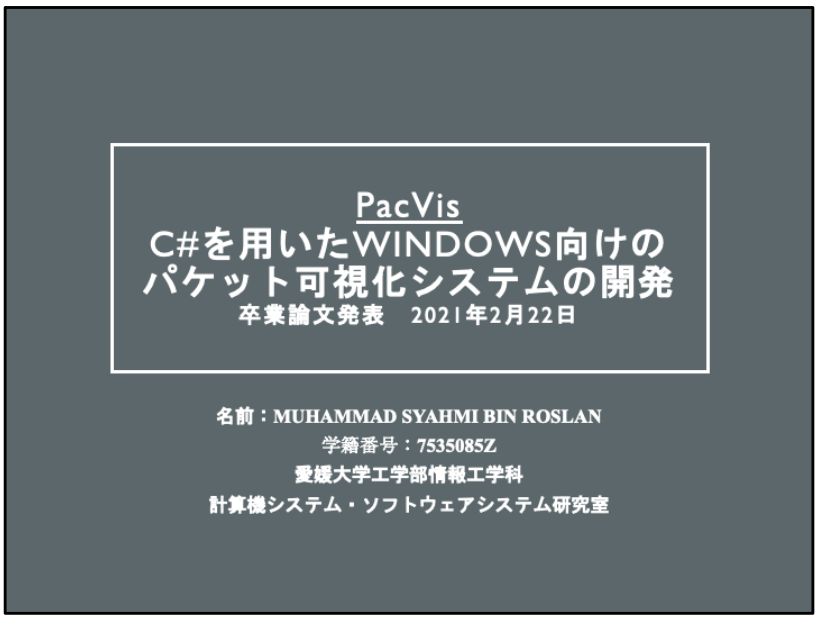
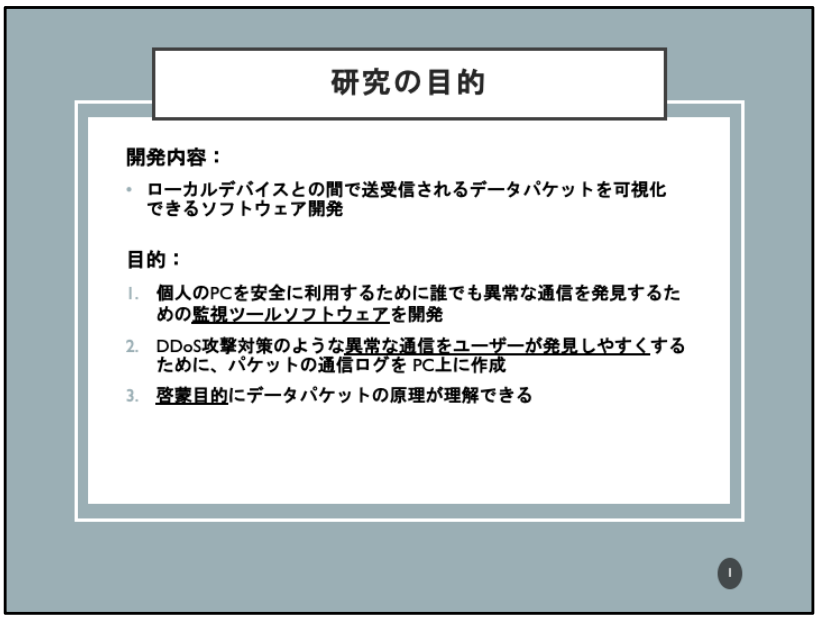


C#を用いたWindows向けのパケット可視化システムの開発と題しまして計算機システム研究室とソフトウェア研究室の合同研究のムハマド が発表します。

どうぞよろしくお願いいたします。



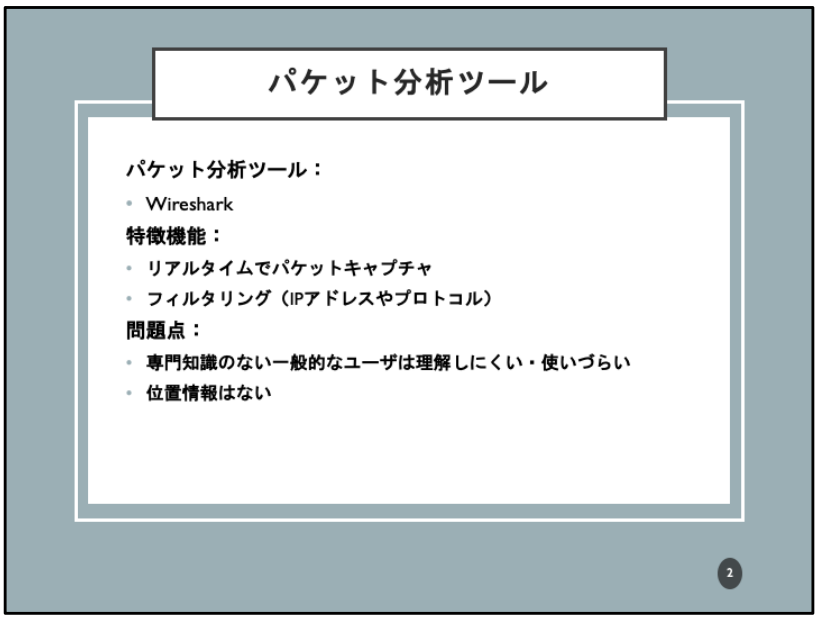
研究内容としては、ローカルデバイスとの間で送受信されるデータパケットを可視化できるソフトウェアを開発します。

この研究の目的としては3つあります。

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ます。

そして、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。

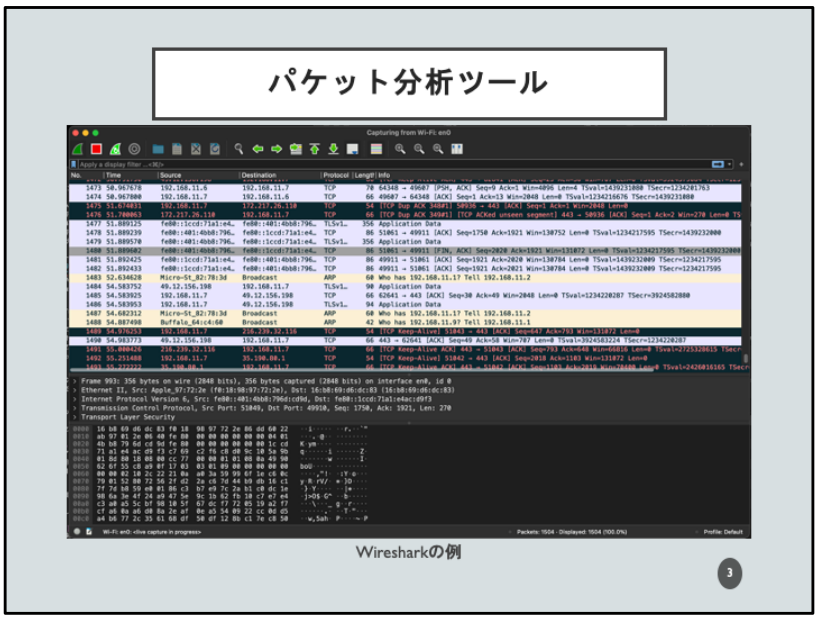


パケットの送受信状況を見る手段としてはWiresharkのようなパケット分析ツールがあります。

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も対応します。

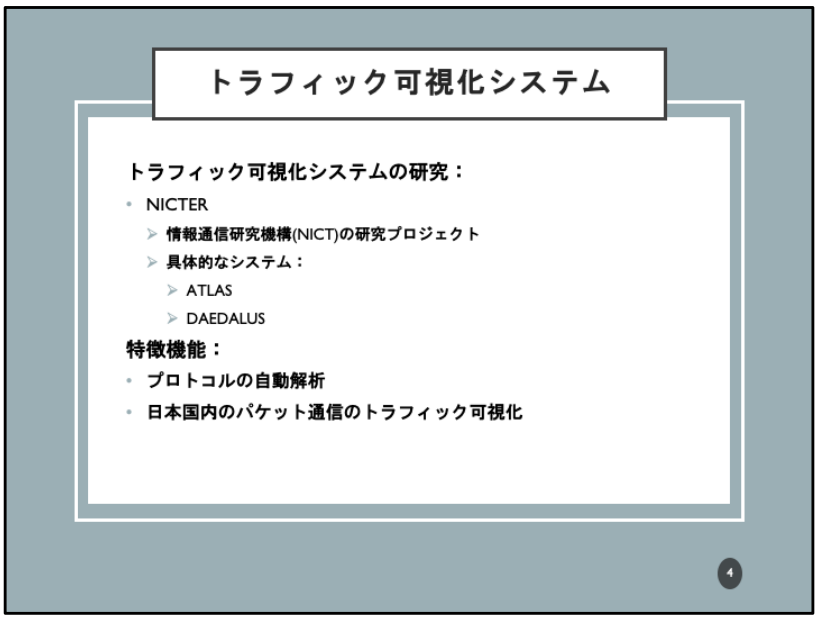
ただし、Wiresharkの場合は、専門知識のない一般的なユーザは理解しにくいし、使いづらいシステムであります。

そして、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。

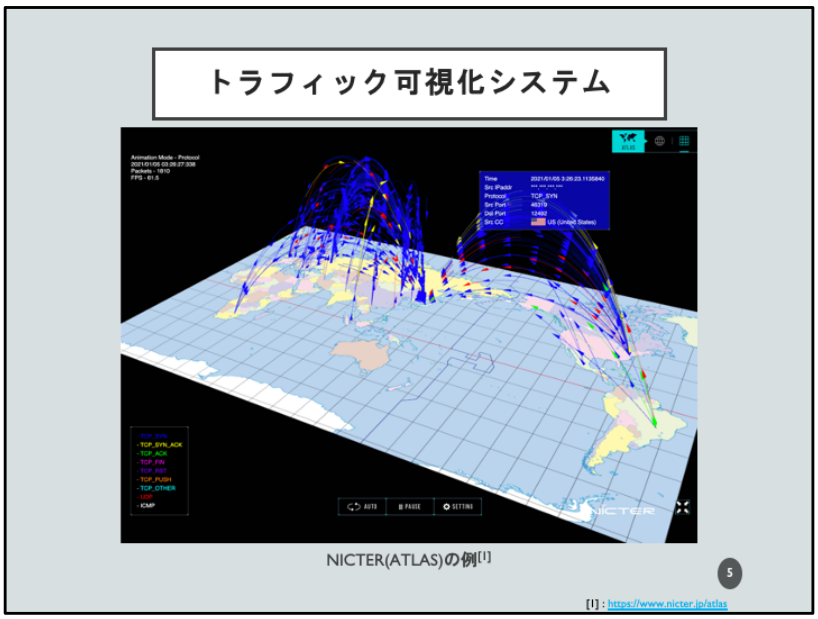
ご覧になっている通り、本システムはわかりにくいシステムであります。



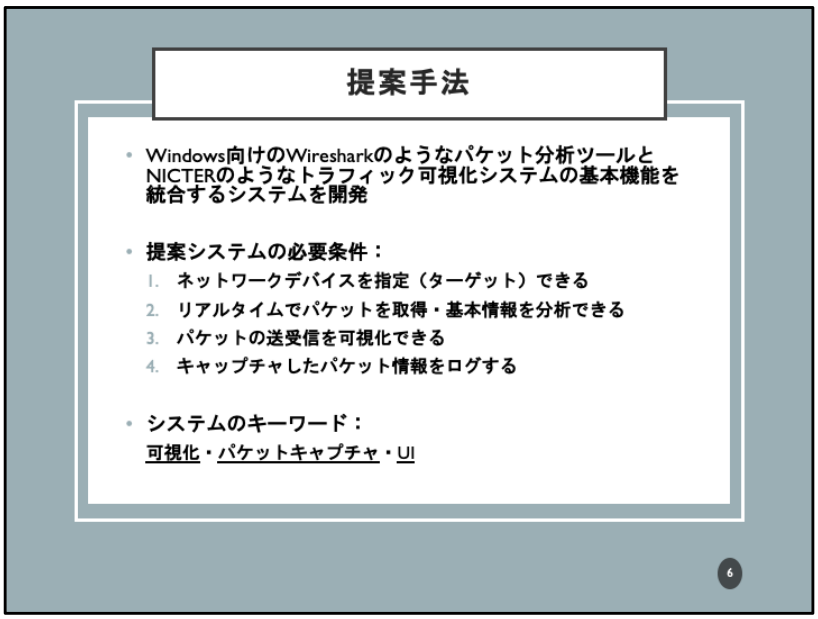
トラフィック可視化システムを紹介しますと、日本内だとNICTERという研究プロジェクトが一般的です。

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を表示します。



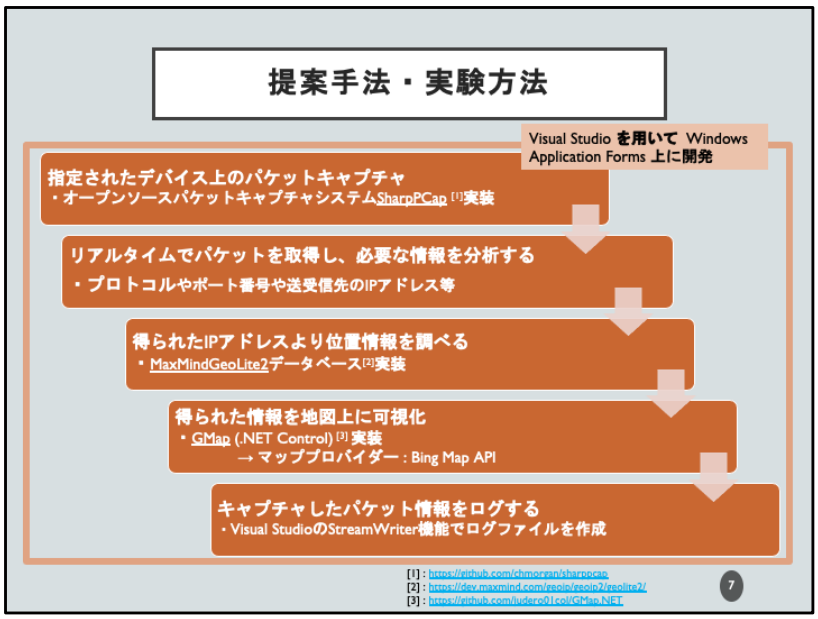
本研究の提案手法に移りますと、Windows向けのWiresharkのようなパケット分析ツールとNICTERのようなトラフィック可視化システムの基本機能を統合するシステムを開発します。

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。

まずは、

1. ネットワークデバイスを指定（ターゲット）できる
2. リアルタイムでパケットを取得・基本情報を分析できる
3. パケットの送受信を可視化できる
4. キャップチャしたパケット情報をログする

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。



実験方法はこのようになります。

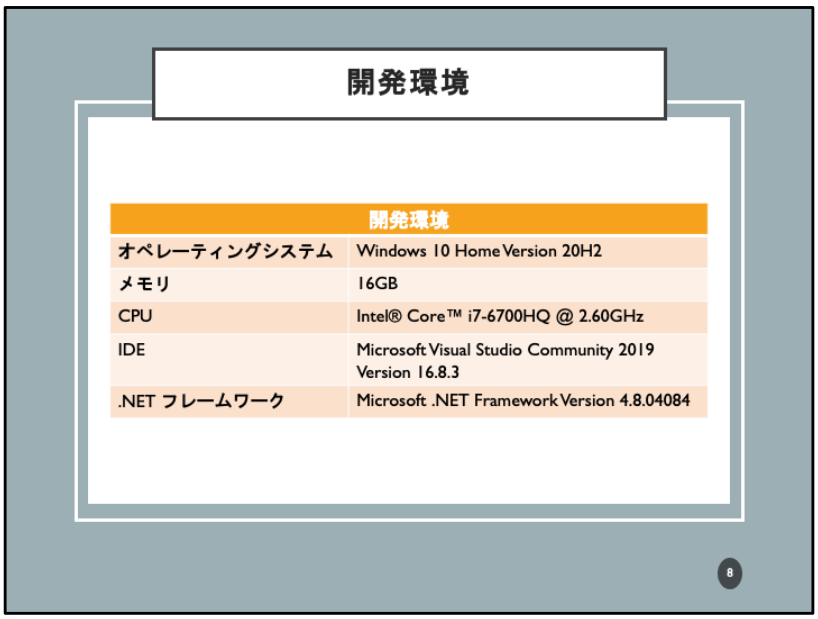
まず、指定されたデバイス上のパケットをキャプチャし、リアルタイムでパケットを取得したら必要な情報を分析します。
この部分はオープンソースパケットキャプチャシステムのSharpPCapを利用しています。

そして、得られたIPアドレスより、MaxMindGeoLite2データベースを使って位置情報を調べます。
得られた位置情報を地図上に可視化をします。可視化を担当するのはGmapの.NET Controlを実装しています。

なお、今回のマッププロバイダーはBing Map APIを使用しています。
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ます。

これらはVisual Studioを用いてWindows Application Forms 上に開発します。

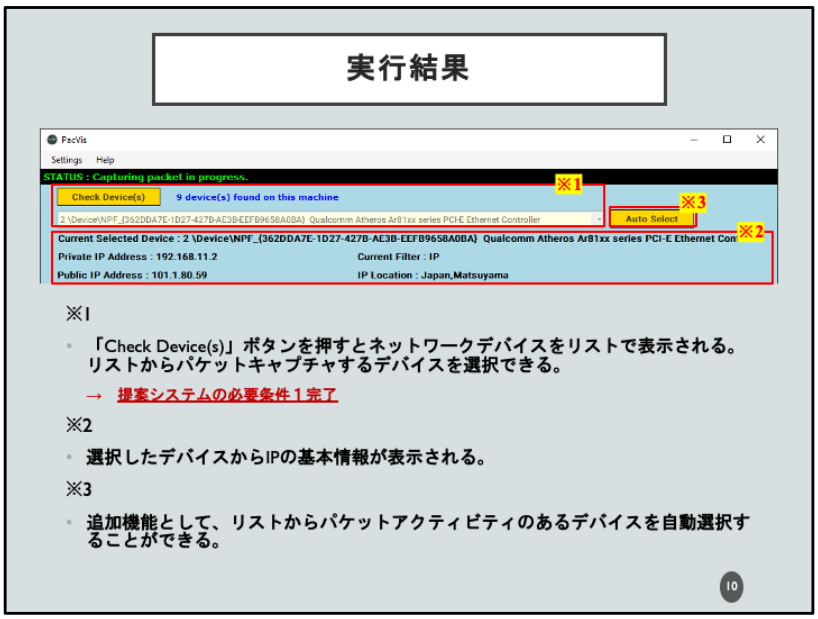


開発環境はこのようになっています。

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という名前をつけます。

PacVisを説明すると、3つの部分に分けることができます。

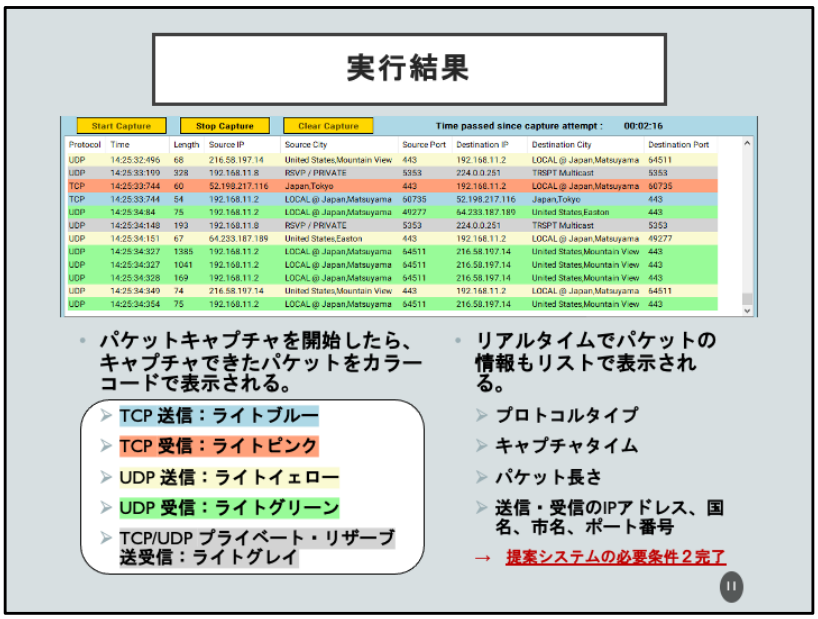


まず、デバイスを管理する部分です。

この部分では３つのポイントがあります。

なお、一番大事なポイントはポイント１となります。

ポイント１は「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、リストからパケットキャプチャするデバイスを選択できる部分は提案システムの必要条件１が達成します。



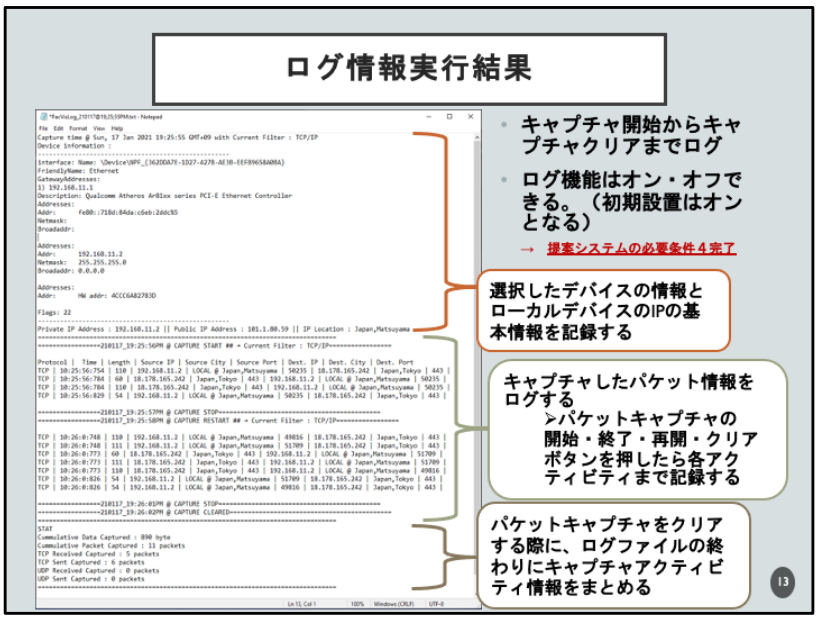
次、パケットキャプチャの分析と情報表示の部分です。

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、提案システムの必要条件２を完了します。

最後に、PacVisの下半分のところで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。

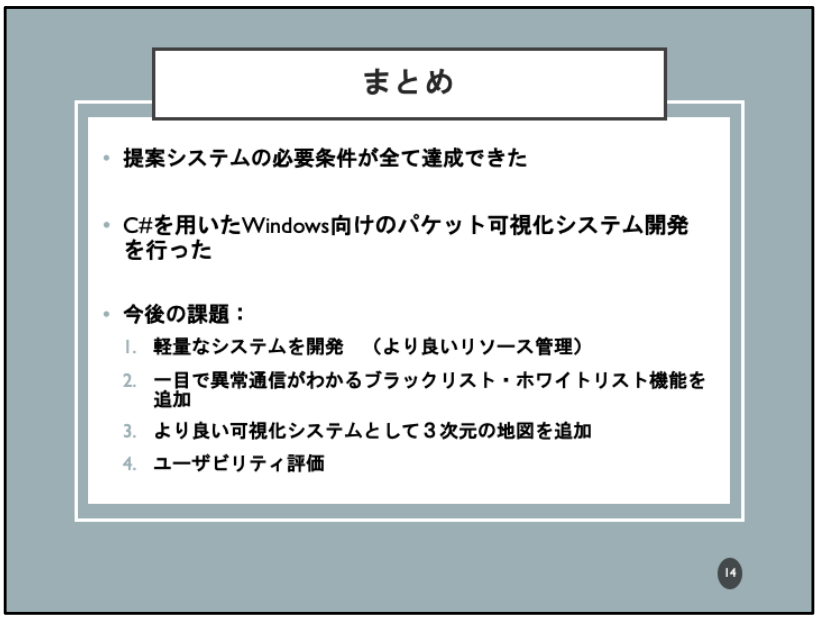
これで提案システム必要条件３が達成しまた。



キャプチャ開始からキャプチャクリアまで全てのアクティビティがログします。

このログ機能で提案システムの必要条件４が完了できました。

なお、ログ機能は設定でオン・オフできます。



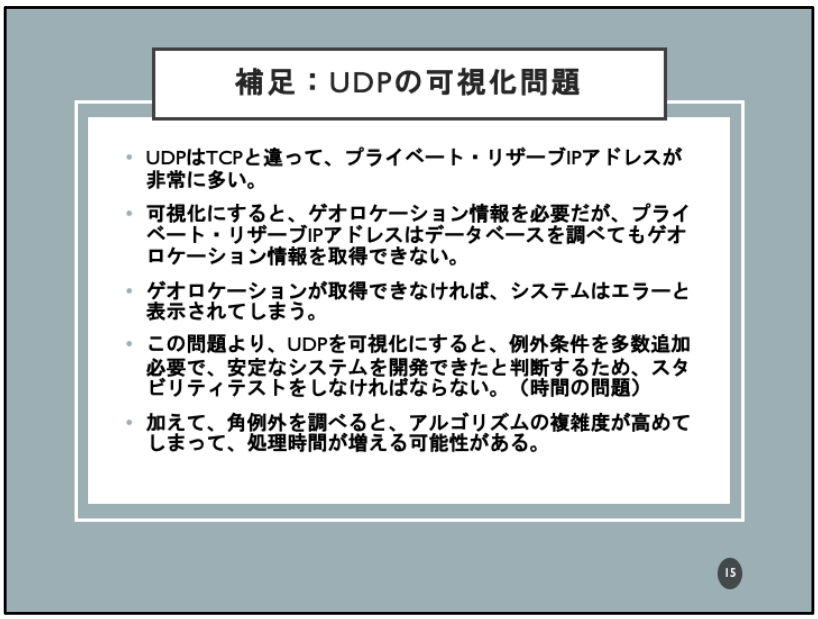
まとめて、提案システムの必要条件が全て達成でき、C#を用いたWindows向けのパケット可視化システム開発が行いました。

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できるかという課題があります。

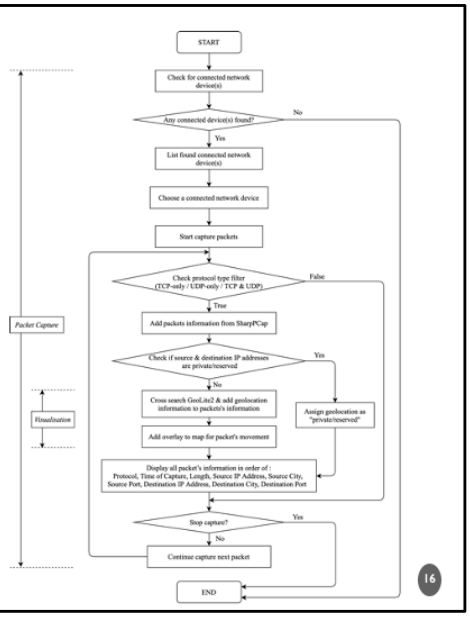
そして、ブラックリスト・ホワイトリスト機能で一目で異常通信がわかるよう、より良い可視化システムとして３次元の地図も追加できたらさらにいいUIができればと思います。

最後に、今回は新型コロナウイルスでユーザビリティ評価ができず、ユーザビリティ評価を実施することも今後の課題になります。

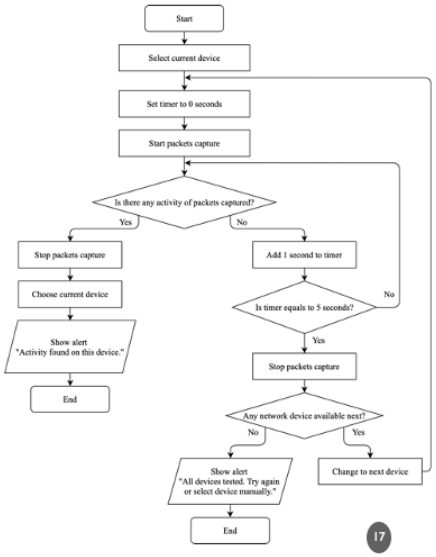
以上で発表終わります。ご静聴ありがとうございました。


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



17