BitVisor内蔵のlwIPで Alkanetログの送信を試みる

立命館大学 システムソフトウェア研究室

山下雄也，明田修平，瀧本栄二，毛利公一

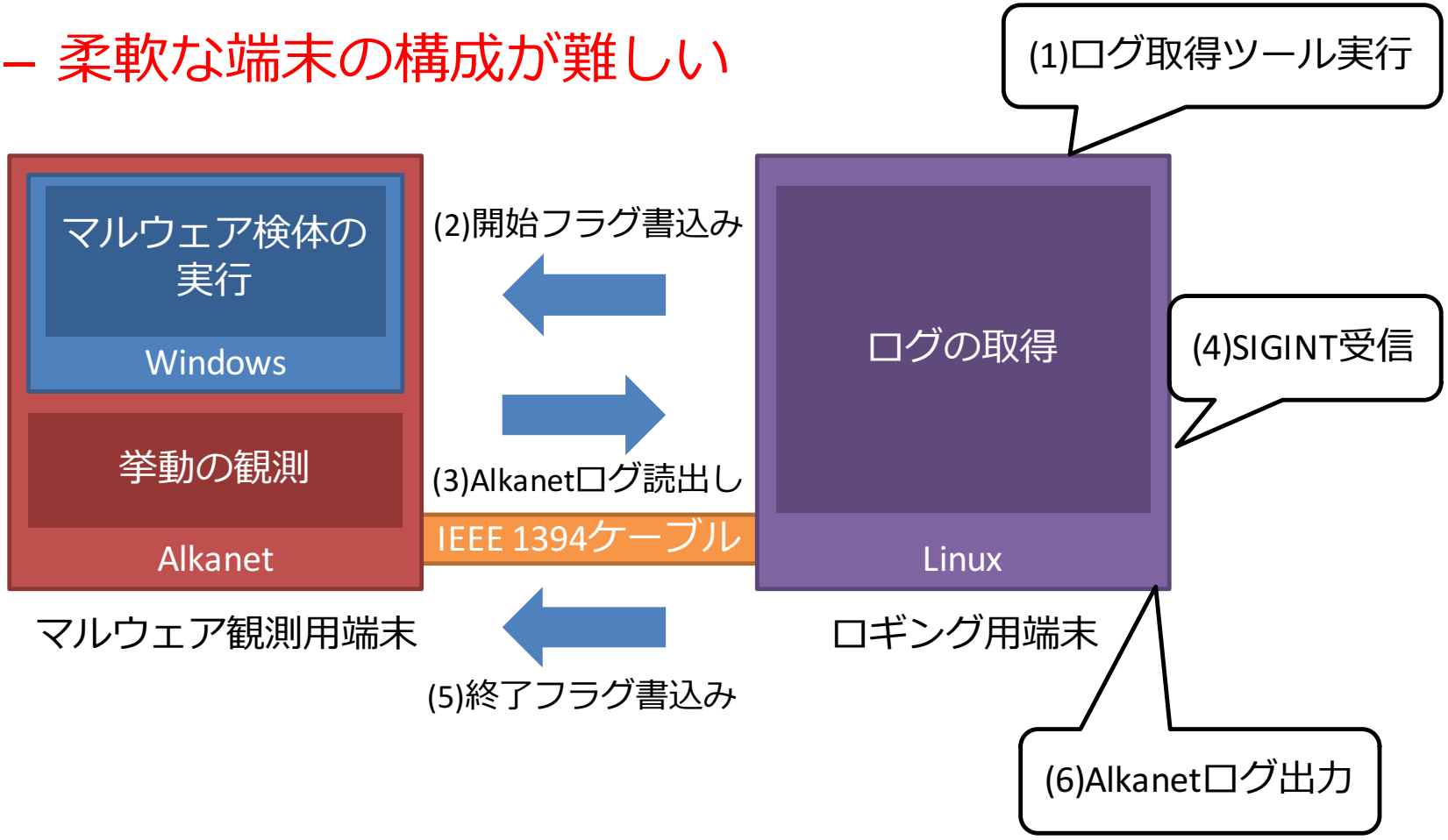
はじめに (1/4)

- 近年，マルウェアが増加しており，
  マルウェアの脅威が問題となっている
  - マルウェアの脅威に対抗するためには，
    多数のマルウェアを迅速に解析する必要がある
- システムコールトレーサAlkanet†
  - Windows上で動作するマルウェアを対象とし，
    マルウェアが発行したシステムコールのログ
    (**Alkanetログ**) を作成
  - VMMであるBitVisorベース
  - アンチデバッグ機能を持つマルウェアでも
    スレッド単位で短時間にトレース可能

†大月勇人, 瀧本栄二, 齋藤彰一, 毛利公一: マルウェア観測のための仮想計算機モニタを用いたシステムコールトレース手法, 情報処理学会論文誌, Vol.55, No.9, pp.2034–2046 (2014).

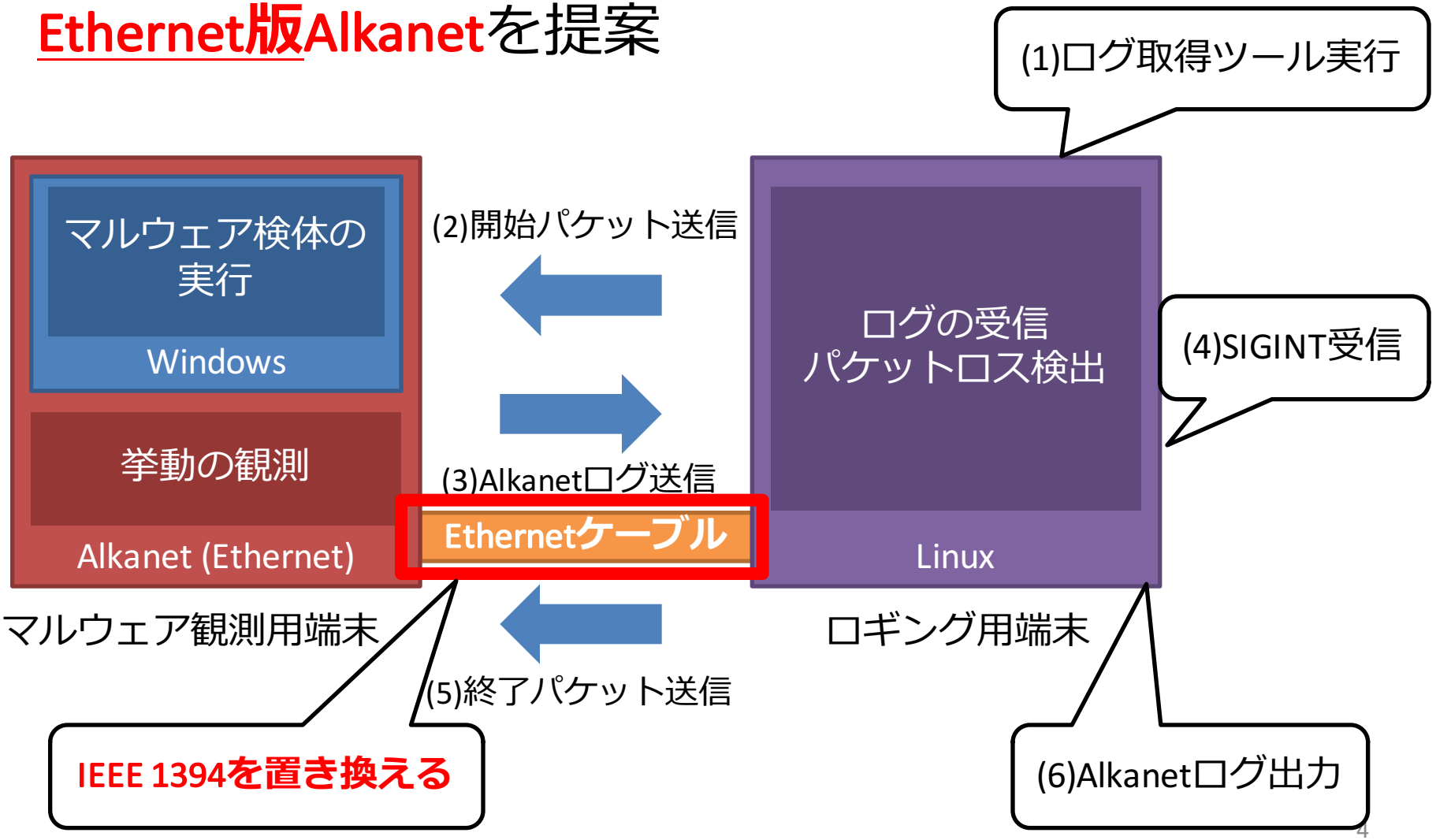
はじめに (2/4)

- Alkanetによるトレースでは，IEEE 1394ケーブルを用いる
- IEEE 1394の問題点
  - 柔軟な端末の構成が難しい



はじめに (3/4)

- 前スライドで述べた問題を解決するため，**Ethernet版Alkanet**を提案



はじめに (4/4)

- 本発表では，Ethernet版Alkanetの開発を進める中で得られた知見について発表する
  - **NICの隠蔽設定や送信に特化したlwIPのチューニングによる性能差**
  - **lwIPを用いて複数のパケットを送信する場合，ARPを考慮する必要がある**

使用するTCP/IPプロトコルスタック

- Ethernet経由でのAlkanetログ送信を実現するためには，TCP/IPプロトコルスタックが必要
  - 現行のBitVisorには，TCP/IPプロトコルスタック**lwIP**が移植されている
  - lwIPを用いて，Ethernet経由でのAlkanetログ送信機能を実装
- lwIP (lightweight IP)
  - 組込みシステムを対象としたTCP/IPプロトコルスタック
  - 設計目標: メモリ等のリソース使用量を可能な限り削減する

各種設定による送信性能の差

送信スループットの計測

- lwIPが，どれほどの送信性能を持っているか調べるため，送信スループットの計測を行った
- 以下の6つを対象に，計測を行った

	プロトコル	NICの隠蔽設定	データサイズ/pkt	送信バッファサイズ
(TCP, ip)	TCP	net=ip	536B	1072B
(tuned TCP, ip)	TCP	net=ip	1460B	65535B
(TCP, ippass)	TCP	net=ippass	536B	1072B
(tuned TCP, ippass)	TCP	net=ippass	1460B	65535B
(UDP, ip)	UDP	net=ip	1472B	
(UDP, ippass)	UDP	net=ippass	1472B	

送信スループットの計測

- lwIPが，どれほどの送信性能を持っているか調べるため，送信スループットの計測を行った
- 以下の6つを対象に，計測を行った

	プロトコル	NICの隠蔽設定	データサイズ/pkt	送信バッファサイズ
(TCP, ip)	TCP	net=ip	536B	1072B
(tuned TCP, ip)	TCP	net=ip	1460B	65535B
(TCP, ippass)	TCP	net=ippass	536B	1072B
(tuned TCP, ippass)	TCP	net=ippass	1460B	65535B
(UDP, ip)	UDP	net=ip	1472B	
(UDP, ippass)	UDP	net=ippass	1472B	

NICの隠蔽設定

- BitVisorは，NICをゲストOSに対して隠蔽する機能を持っている
  - **net=ip**: NICを隠蔽する
  - **net=ippass**: NICを隠蔽しない
- NICを隠蔽するか否かが，
  送信性能にどの程度影響を与えるのか知りたい
  - NICの隠蔽設定を計測対象に加えた

送信スループットの計測

- lwIPが，どれほどの送信性能を持っているか調べるため，送信スループットの計測を行った
- 以下の6つを対象に，計測を行った

	プロトコル	NICの隠蔽設定	データサイズ/pkt	送信バッファサイズ
(TCP, ip)	TCP	net=ip	536B	1072B
(tuned TCP, ip)	TCP	net=ip	1460B	65535B
(TCP, ippass)	TCP	net=ippass	536B	1072B
(tuned TCP, ippass)	TCP	net=ippass	1460B	65535B
(UDP, ip)	UDP	net=ip	1472B	
(UDP, ippass)	UDP	net=ippass	1472B	

TCPのチューニング

- lwIPは，チューニング用の#define定数マクロを多数用意している
  - チューニングを行うことで，どの程度高速になるのか知りたい
- 以下の2つのチューニングを行ったtuned TCPを計測対象に加えた
  - MSSを536Bから1460Bに変更 (TCP_MSS)
  - TCPが持つ送信バッファのサイズを1072Bから65535Bに変更 (TCP_SND_BUF, TCP_SND_QUEUELEN, MEMP_NUM_TCP_SEG)

送信スループットの計測

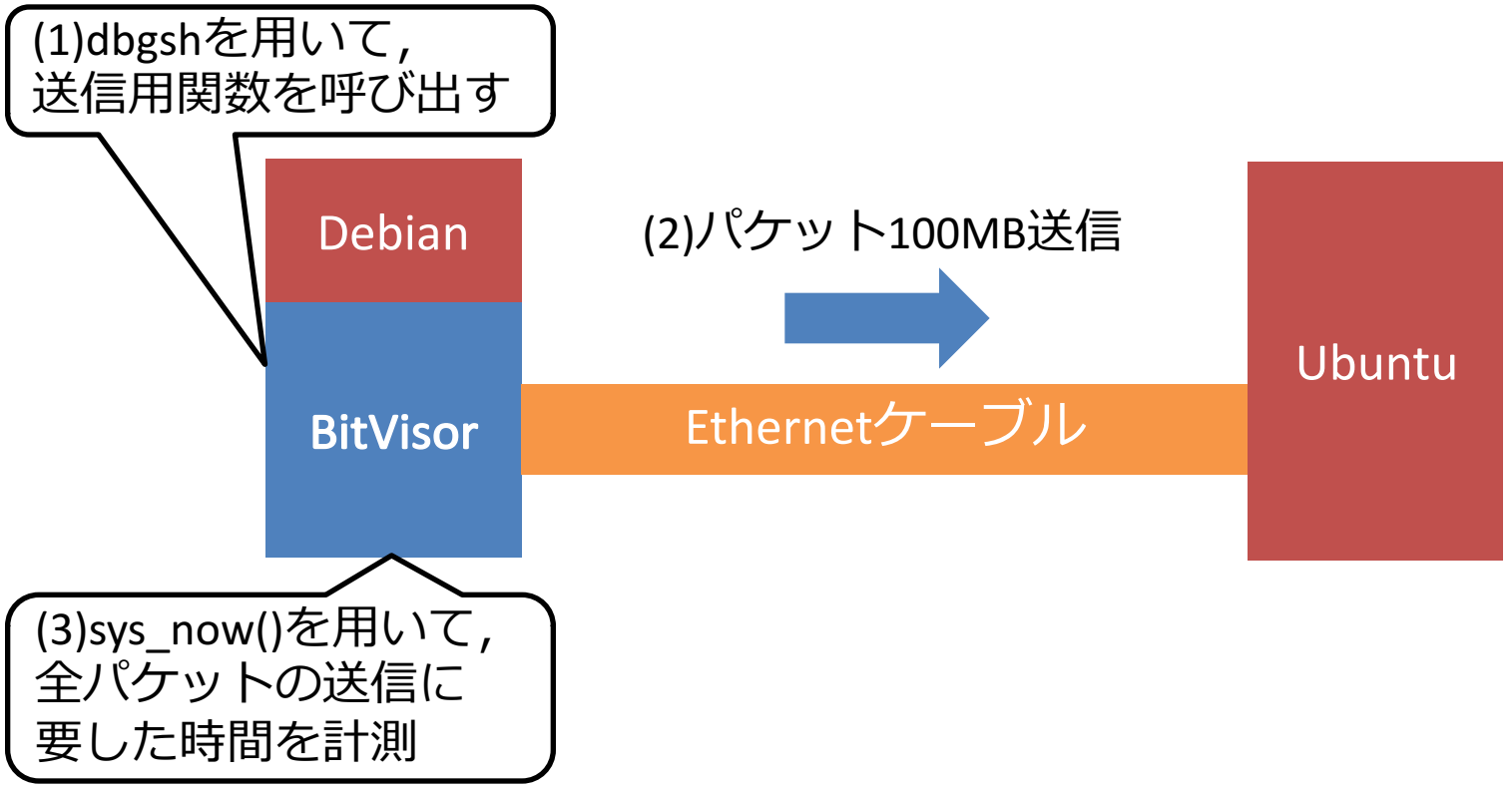
- lwIPが，どれほどの送信性能を持っているか調べるため，送信スループットの計測を行った
- 以下の6つを対象に，計測を行った

	プロトコル	NICの隠蔽設定	データサイズ/pkt	送信バッファサイズ
(TCP, ip)	TCP	net=ip	536B	1072B
(tuned TCP, ip)	TCP	net=ip	1460B	65535B
(TCP, ippass)	TCP	net=ippass	536B	1072B
(tuned TCP, ippass)	TCP	net=ippass	1460B	65535B
(UDP, ip)	UDP	net=ip	1472B	
(UDP, ippass)	UDP	net=ippass	1472B	

計測環境

項目	値
VMM	BitVisor 1.4
ゲストOS	Debian 8.6 (Linux 3.16.0-4.amd64)
CPU	Intel (R) Core(TM) i5-2320 CPU @ 3.00GHz
メモリ	4GB
NIC	Intel Corporation PRO/1000 PT Dual Port Server Adapter (1Gbps)
1パケットあたりのデータサイズ	1460B (TCP)，1472B (UDP)
1実験あたりの合計送信データサイズ	100MB
実験回数	5回

計測方法



- 送信した合計バイト数と送信に要した時間から送信スループットを算出
- 5回計測し，その平均値を結果とする

計測結果と考察

(TCP, ip)	138.3Kbps
(tuned TCP, ip)	21.2Mbps
(TCP, ippass)	33.8Mbps
(tuned TCP, ippass)	874Mbps
(UDP, ip)	953Mbps
(UDP, ippass)	981Mbps

- 特に，TCPの送信スループットが極端に低下する
  - TCPは，ウィンドウサイズ以上のデータを送信する場合，ACKを待つ必要がある
  - NICを隠蔽することで，VMに割込みが入らない
  - ACKの受信処理を円滑に行えないことが原因

計測結果と考察

(TCP, ip)	138.3Kbps
(tuned TCP, ip)	21.2Mbps
(TCP, ippass)	33.8Mbps
(tuned TCP, ippass)	874Mbps
(UDP, ip)	953Mbps
(UDP, ippass)	981Mbps

153倍

26倍

- チューニングに関する参考サイト
  - Tuning TCP | lwIP Wiki | Fandom powered by Wikia (http://lwip.wikia.com/wiki/Tuning_TCP)
  - Maximizing throughput | lwIP Wiki | Fandom powered by Wikia (http://lwip.wikia.com/wiki/Maximizing_throughput)

計測結果と考察

(TCP, ip)	138.3Kbps
(tuned TCP, ip)	21.2Mbps
(TCP, ippass)	33.8Mbps
(tuned TCP, ippass)	874Mbps
(UDP, ip)	953Mbps
(UDP, ippass)	981Mbps

- Ethernet版Alkanetでは，(UDP, ip)を採用
  - ロギング用端末への攻撃の対策
  - 膨大なログを素早く送信する必要がある
  - 有線なので，パケットロスは滅多に発生しない

ARPに関する問題

ARPに関する問題

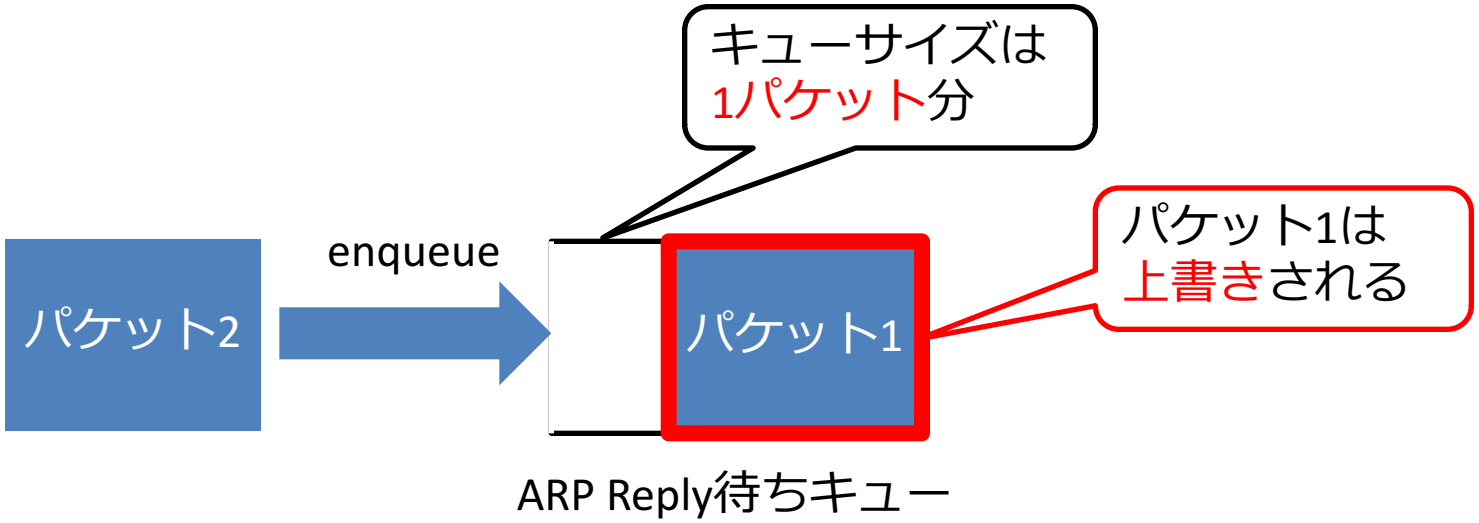
- ARPによるアドレス解決が完了する前に，複数のパケットを送信しようとするとパケットの破棄が発生する
  - 複数のパケットを送信する場合，ARPを考慮する必要がある
- 問題の原因と対策について述べる

lwIPの送信処理の流れ

- lwIPは，以下の手順で送信処理を行う

1. TCP or UDPヘッダを作成
2. IPヘッダを作成
3. ARPテーブルを検索
   - 該当するエントリあり
     4. Ethernetヘッダを作成してパケットを送信
   - 該当するエントリなし
     4. ARP Reply待ちキュー (ARPエントリ毎に存在) にパケットを一時的に格納
     5. ARP Requestを送信
     6. ARP Reply受信後，4で格納したパケットを送信

ARPに関する問題の原因



- nパケット送信した場合，
  1番目からn-1番目までのパケットは破棄される
- Ethernet版Alkanetでは問題にならない
  - 起動パケットを受信した段階で，アドレス解決が完了しているため

複数のTCPパケット送信方法

- tcp_connect()を用いることで，ARPアドレス解決や3ウェイハンドシェイクと同期が取れる
- err_t tcp_connect(struct tcp_pcb *pcb, ip_addr_t *ipaddr,
  u16_t port,
  tcp_connected_fn connected)
  - SYNパケットを送信する (ARP Requestも送信)
  - connectedには，コネクション確立後に呼び出されるコールバック関数を指定可能
- connectedにデータ送信用関数を指定することで，複数のTCPパケットを送信できる
  - コネクション確立後には，ARPアドレス解決が完了しているため

複数のUDPパケット送信方法

- TCPのように，コールバック関数で同期を取ることができない
- ARPに関して何かしらの対応は必要
  - **1パケット目と2パケット目の間に待機処理を入れる**
  - **lwIPを拡張し，ARP Reply受信時にコールバック関数が呼ばれるようにする**
  - **ARP Reply待ちキューのサイズを増やす**
- ARP Reply待ちキューのサイズの増やし方
  - ARP_QUEUEINGを1に設定
  - MEMP_NUM_ARP_QUEUEにサイズを設定 (パケット単位)

Ethernet版Alkanetの現状

- 基本的な機能については実装完了
  - Alkanetログ送信機能
  - 起動/終了パケット受信機能
  - Ethernet版ログ取得ツール
- 今後は，Ethernet版Alkanet上でマルウェアを動作させ，IEEE 1394版Alkanetと同様のログを取得できるか確認

おわりに

- Ethernet版Alkanetの開発を進める中で
  得られた知見について発表
  - NICの隠蔽設定やlwIPのチューニングによる送信性能の差
    - NICを隠蔽すると，特にTCPの送信スループットが
      極端に低下する
    - lwIPが提供するチューニング用定数を使用することで，
      送信スループットの向上が可能
  - lwIPを用いて複数のパケットを送信する場合，
    ARPを考慮する必要がある
    - TCPでは，コールバック関数を用いて解決可能
    - UDPでは，これといった解決策が存在しない