

災害時における SNS 通信の優先制御システムの設計 Design of Priority-Based Control System for SNS Traffic in Disaster Situations

田中 秀征¹⁾ 佐藤 寧洋¹⁾
Hideyuki Tanaka Yasuhiro Sato

1 はじめに

日本は地震や豪雨、台風など自然災害が発生しやすい国土であり、毎年のように広範囲にわたる災害が発生している。災害発生時には災害情報や避難情報といった刻々と変化する情報を、適時かつ多角的に入手することが被害拡大の防止や命を守るうえで大変重要である。近年ではテレビやラジオなどの一方向的な情報伝達手段に加えて、Social Networking Service (SNS) がリアルタイム性の高い情報共有手段として活用されている。SNS は災害時において、ユーザが即時に発信・共有だけでなく行政や自治体、報道機関等の公的機関においても活用されている。大規模な地震や豪雨災害の発生時には、X や LINE 公式アカウントを通じて自治体が避難情報や支援物資の情報を発信し、住民との迅速な情報共有を図っている事例が多数報告されている。しかし、災害発生時やその他の緊急時には多くの人々が同時多発的にネットワークを利用するため、平常時と比較してネットワークが輻輳状態に陥やすく、SNS への投稿・閲覧が通常通り行えないだけでなく、安否確認や避難情報の取得、消防・警察・医療機関との連絡といった、迅速な対応を要する通信が遅延・途絶し、被害の拡大や対応の遅れを引き起こすおそれがある。相次ぐ自然災害の発生から、被害の拡大を事前に防止することは困難であるが、災害発生後の被害軽減や迅速な対応を可能にするにはあらかじめ通信環境の整備が必要である。

本稿では、そのような状況に陥ったネットワークにおいても、災害救助などに係わる重要な通信を優先的に保護することで、必要とされている通信を維持する方法を検討する。具体的には、柔軟なネットワーク制御が可能である OpenFlow 技術を利用し、気象庁から発出される災害情報をもとに被災地を特定、その地域からの SNS トラフィックに対して優先度といわれる独自の指標を定め、その数値に応じてよりスムーズな情報配信を可能とする優先制御システムの提案を目指す。

2 優先度

本システムでは、被災地からの SNS トラフィックがもっとも優先されるように優先度を設定し、その値によって通信を制御する。各通信フローの優先度は、表 1 の送信元端末の位置情報と表 2 の SNS 別の優先順位の加算によって算出する。SNS 別の優先順位は、総務省 [1] が公表している各サービスの利用者数に基づいた順位とした。優先度が高く算出されたパケット（フロー）に対してはリソースを多く割り当て一方で、通常の Web 閲覧やリソースを多く占領するオンラインゲームなどの通信は緊急性が低いケースが多く、優先度を下げることでネットワーク全体のリソースを確保する。また被災地以外からの通信も考慮してネットワーク

1) 大阪電気通信大学大学院工学研究科

Graduate School of Engineering, Osaka Electro-Communication University

表 1: 端末の位置情報

地域	スコア
被災地	+1
非被災地	+0

表 2: SNS 別の優先順位

順位	SNS	スコア
1	LINE	+4
2	Instagram	+3
3	X	+2
4	Facebook	+1
5	上記以外	+0

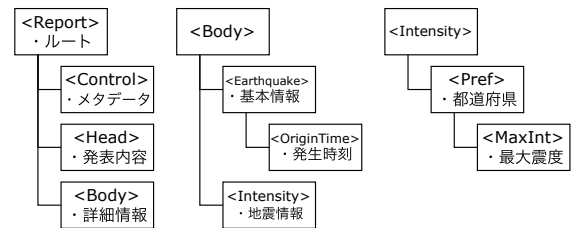


図 1: 気象庁 XML の階層構造

制御を行うため、本制御を行う中継ルータはアクセスネットワークよりも、よりトラフィックを集中して制御を行えるサーバ側に近いコアネットワークに配置することが望ましいと考えている。

3 優先制御に用いるデータの取得方法

本項では、優先制御をするにあたっての各種データの取得方法について概説する。

3.1 災害情報の取得

災害情報の取得には気象庁が随時配信している気象庁防災情報 XML [2] を利用し、都道府県ごとの最大震度を自動で取得するプログラムを Python で実装した。図 1 に地震情報に関する XML の構造を示す。XML はルート要素 <Report> をルートとして、配信元等を示すメタデータである <Control>、発表内容を示す <Head>、地震の詳細情報を示す <Body> の 3 つのセクションで構成されている。地震の震源情報や各地の震度は <Body> 以下に階層的に格納されており、これを解析することで地域ごとの震度情報を取得する。図 2 に動作検証のために某日の地震情報をプログラムにより取得した一例を示す。運用時には取得した情報をトリガとしてネットワークを動的に切り替える設計を想定している。この XML は随時配信されているが、ネットワークへの負荷や解析する端末の負荷を考慮し 10 分間隔で監視を行う。平常時は特定のトラフィックに対する優先的な制御は行わないが、緊急地震速報が配信された場合と警戒レベル 4 以上（従来の避難指示に相当）が配信された場合に限り自動的に本システムを有効化する。

3.2 発信者位置情報の推定

災害時には停電や固定回線の障害により、発信者の多くは携帯アクセス回線からインターネットに接続していることが想定される。本稿では発信者が携帯回線からインターネットにアクセスしていることを前提に、回線通信事業者から割り当てられている IP アドレスをもと

地震発生時刻: 2025-06-07T14:40:00+09:00
 都道府県: 青森県, 震度: 2
 都道府県: 北海道, 震度: 1
 都道府県: 岩手県, 震度: 1
 地震発生時刻: 2025-06-07T12:40:00+09:00
 都道府県: 広島県, 震度: 1
 地震発生時刻: 2025-06-07T09:56:00+09:00
 都道府県: 熊本県, 震度: 1

図2: 取得した地震情報

に推定を行う。位置推定には、MaxMind 社が提供する GeoLite2 [3] のデータベースから端末の位置情報を推定する。実際に、大阪、兵庫、奈良の主要駅周辺からスマートフォンの携帯回線を使用して実験用サーバにアクセスし、ログから取得した送信元 IP アドレスをもとに、上記の方法を利用することで、市区町村レベルまで正しく位置情報が推定できた。

3.3 SNS パケットの識別

SNS トラフィックにおける通信フローはエンド間で暗号化されているため、データパケットからトラフィックの種類を識別することが困難であることから、パケットのデータ部分によらない識別手法が必要とされる。本稿では中継ルータにおいて通信開始前に発生する DNS クエリレスポンスのパケットから、ドメイン名とその IP アドレスを取得する。DNS レスポンスは暗号化されていない場合が多く、サービス名を含むドメイン名の取得が可能であることから、SNS 関連トラフィックの識別に適していると考えられる。取得したドメイン名には各 SNS サービス名が含まれているため、その IP アドレス宛のトラフィックを SNS サービスに係わるフローと判断する。近年では、リバースプロキシなどの利用が一般的になっており、IP アドレスがサービスと対応していないケースもあり得るが、これによる通信制御における影響については今後の課題とする。また、DNS クエリをキャプチャすることについても、通信の秘匿性に係るセキュリティ的な懸念があるため、事前に SNS サービスの IP アドレスを取得しておくことなども別途検討している。

4 実験環境

図3にネットワークエミュレータである Mininet を用いたネットワーク構成を示す。本環境は被災地の端末を h1、被災地以外の端末を h2 と想定し、h3 を SNS サーバ、h4 をその他のサービスの Web サーバ、h5 を DNS サーバとしている。これらの端末およびサーバは中継ルータである OpenFlow スイッチで集線されている。OpenFlow スイッチは Mininet 上に構成した仮想スイッチであり、OpenvSwitch を用いて動作している。スイッチのプロトコルバージョンは OpenFlow 1.3 に設定し、OpenFlow コントローラである Ryu を用いてスイッチを制御する。

5 DNS レスポンスパケットの取得

3.3に示したとおり、本システムでは DNS レスポンスパケットの取得により SNS を識別する。DNS レスポンスパケットの取得には OpenFlow の制御機能の1つである Packet-In メッセージを利用する。Packet-In メッセージは OpenFlow スイッチが特定のパケットにマッチした場合、スイッチからコントローラへ送信される制御メッセージである。スイッチ上で送信元ポート番号が 53 番かつ UDP パケットを対象にアクションを設定し、

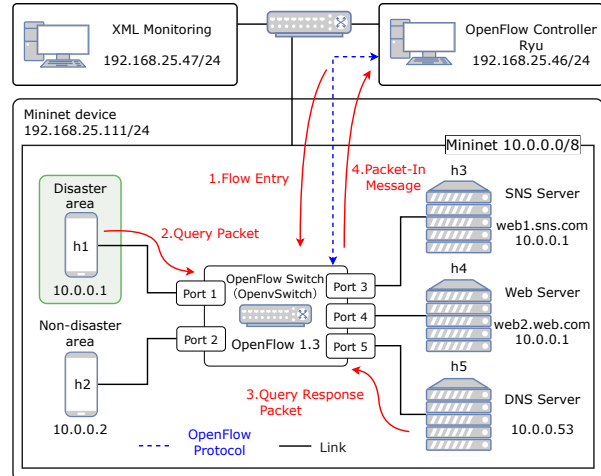


図3: 実験環境

スイッチから Ryu コントローラに Packet-In メッセージを送信するようにフローエントリを作成した。実際に作成したフローエントリをコード1に示す。Ryu コントローラでは受信した Packet-In メッセージから DNS レスポンスの Answer セクションからドメイン名および宛先サーバの IP アドレスを取得する。本システムでは、この IP アドレスが特定の SNS サービスに対応しているかを照合し、対応している場合はそのトラフィックに対して優先度を決定し、それに応じたキューに格納するようなフローエントリを作成することで、優先度に基づいた制御を行う。

コード1: 作成したフローエントリ

```

1  ### マッチ条件をUDPかつ送信元ポート番号53番
2  match_dns = parser.OFPMatch(
3      eth_type=0x0800, ip_proto=17,
4      udp_src=53)
5
6  ### Packet-Inを定義
7  actions_dns = [
8      parser.OFPACTIONOutput(
9          ofproto.OFPP_CONTROLLER,
10         ofproto.OFPCML_NO_BUFFER),
11         parser.OFPACTIONOutput(ofproto.OFPP_FLOOD)]
12
13  ### フローエントリを追加
14  self.add_flow(
15      datapath, priority=100,
16      match=match_dns, actions=actions_dns)

```

6 おわりに

災害時における SNS 通信の優先制御システムを提案し、OpenFlow を使った実装に向けた提案手法を示した。今後は本システムの実装を目指し、より実環境に近い大規模なネットワーク構成による性能評価や優先度の決定に用いる数値の最適値について検討する。

参考文献

- [1] 総務省情報通信政策研究所: 令和4年度情報通信メディアの利用時間と情報行動に関する調査報告書, https://www.soumu.go.jp/main_content/000887659.pdf.
- [2] 気象庁: 気象庁防災情報 XML フォーマット, <https://xml.kishou.go.jp/xmlpull.html>.
- [3] MAXMIND: GeoLite Database, <https://dev.maxmind.com/geoip/geolite2-free-geolocation-data>.