

# Programmable Overlay Network の開発

## —Overlay Network を用いた共通基盤構築ミドルウェア—

### 1. 背景

現在, Overlay Network を利用した分散アプリケーションの開発が盛んに行われている. Overlay Network を利用したアプリケーションでは, ルーティングなどにバグがあった場合, アプリケーションの動作を止めた上で Overlay Network を再構築する必要があるため, 分散アプリケーションの開発・管理・運用に手間がかかるという問題が存在する.

### 2. 目的

本プロジェクトでは, Overlay Network を動的に制御可能にするミドルウェアの開発を目的とする. 本ミドルウェアの特徴として, RAT-message (Routing And Topology-message) と呼ぶメッセージの伝搬によって, Overlay Network の動作を自由に変更する事が可能となっている. これにより, 従来の分散アプリケーションで発生していた, 不正トラフィックの発生や予期せぬバグが発生した場合でも, RAT-message に修正点を記述するだけで, 容易に Overlay Network の管理を行う事が可能となる.

### 3. 開発の内容

本プロジェクトでは RAT-message の伝搬を高信頼で達成する基盤の Overlay Network(以下 InfraOverlay Network)と, その上で制御が可能な Overlay Network(以下 Service Overlay Network)の開発を行った. 図1に概要図を記す. なお本ミドルウェアを利用する場合には JRE6.0 のインストールが必要となる.

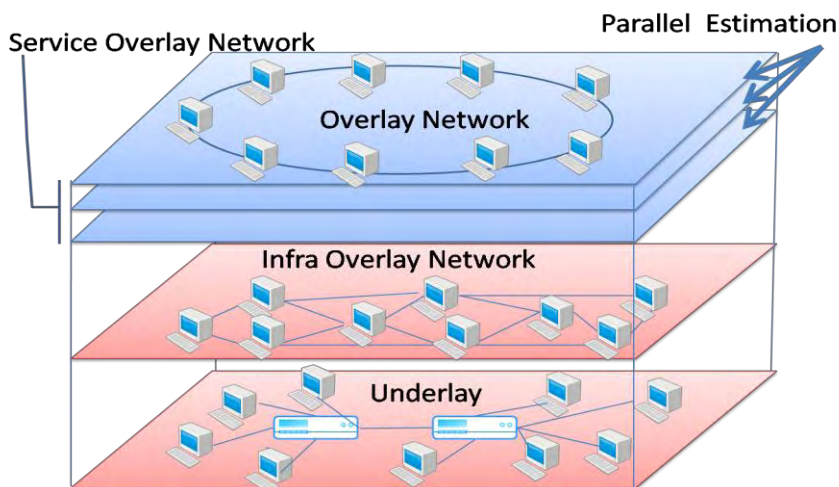


図1 概要図

### 3. 1 InfraOverlay Network

InfraOverlay Network では RAT-message を全てのノードに対して高信頼に伝搬する役割を持っている。そのため、ノードの Overlay Network への参加や離脱を検知するため、制御メッセージを定期的送信している。具体的には、全てのノードが互いに制御メッセージをやり取りした場合、トラフィック量が大きくなるため、各ノードに ID 番号を振り分け、自分の ID の隣に位置する ID を持つノードに対してだけ制御メッセージのやり取りを行うように構造化を行った。この ID による構造化により、木構造を利用したマルチキャストを行う事が出来るため、RAT-message の効率的な伝搬も可能になっている。

### 3. 2 Service Overlay Network

Service Overlay Network では一つの Overlay Network だけでなく、複数の Overlay Network を同時に共存させ、制御・管理を行う事が可能となっている。また、それらの複数の Overlay Network を独立して動作させる事が可能となっている。なお、Overlay Network の追加や制御・管理には、RAT-message を用いる。RAT-message には表 1 に記す API を実装した Java Code を記述する必要がある。この Java Code を本ミドルウェアの管理アプリケーションを用いて、GUI 上にロードし、Activate するだけで、簡単に RAT-message の内容を反映させる事が可能となる。

表 1 API

API	API Name	説明
CoreAPI	void makeRoutingTable(InfraManager)	・ InfraOverlay Network のノード情報を利用しルーティングテーブルを作成
	ID getNextRoute(ID target)	・target の ID に対し、ホップ先のノード ID を返す
SubAPI	boolean copeWithNodeJoin(ID joinNodeID)	・ノードの参加を検知した際の、動作を記述
	boolean copeWithNodeLost(ID lostNodeID)	・ノードの離脱を検知した際の、動作を記述

### 3. 3 動作検証

Service Overlay Network 上で Overlay Network を複数動作させ、それらの独立した動作が可能になっている。これらの各 Overlay Network の動作を検証するために、管理アプリケーションを作成した。検証項目として、各 Overlay Network 上で発生するトラフィック量(byte), メッセージ数, プロトコルメッセージを取得する事が可

能となっている。また、ノード間のメッセージの送信状態を可視化することも可能になっている。図 2、図 3 に検証項目の例と、可視化のスナップショットを記す。

NodeID	NAT	NodeID	From	Cmd	PacketSize	Time
2955361079461864949857383524	false	-463989221200	-4459260420841	searchContents	4286	14:24:38
3246905730384353402827996264	false	-463989221200	-4459260420841	searchContents	4286	14:24:38
3713560123587239601378710774	false	-463989221200	-4459260420841	searchContents	4514	14:25:37
663870959543554884614729269	false	-463989221200	-4459260420841	searchContents	3146	14:25:37
28083369246064942513531337	false	-463989221200	-4459260420841	searchContents	3089	14:25:37
26219323682392643994972422	false	-463989221200	-4459260420841	searchContents	3203	14:25:37
463988922120091077915615394	false	-463989221200	-4459260420841	searchContents	4058	14:25:37
47243650643188443084873517	false	-463989221200	-4459260420841	searchContents	2918	14:25:37
445926042084104773912099679	false	-463989221200	-4459260420841	searchContents	2633	14:25:37
621018232924389295129916485	false	-463989221200	-4459260420841	searchContents	3317	14:25:37
51833586940668574841118248	false	-463989221200	-4459260420841	searchContents	2405	14:25:37
359790089198934043743091786	false	-463989221200	-4459260420841	searchContents	4343	14:25:37
9274948522467355457870413833	false	-463989221200	-4459260420841	searchContents	4628	14:25:37
702263641118170747913640179	false	-463989221200	-4459260420841	searchContents	4400	14:25:37
540268938921660701476733159	false	-463989221200	-4459260420841	searchContents	3431	14:25:37
474599188133805353995681847	false	-463989221200	-4459260420841	searchContents	3887	14:25:37
1358401862350562004751078446	false	-463989221200	-4459260420841	searchContents	2975	14:25:37
551471427375159698302787202	false	-463989221200	-4459260420841	searchContents	4571	14:25:37
529276495949618943441467936	false	-463989221200	-4459260420841	searchContents	2690	14:25:37
210150381166718640436741566	false	-463989221200	-4459260420841	searchContents	4457	14:25:37
372997628127197291434218339	false	-463989221200	-4459260420841	searchContents	3032	14:25:37
508774292573989814298631309	false	-463989221200	-4459260420841	searchContents	3488	14:25:37
176287658577013784020088130	false	-463989221200	-4459260420841	searchContents	3559	14:25:37
417722356257819167799079540	false	-463989221200	-4459260420841	searchContents	2519	14:25:37
2291029754065307845051794665	false	-463989221200	-4459260420841	searchContents	2291	14:25:37
3091541082990028151247159449	false	-463989221200	-4459260420841	searchContents	2348	14:25:37
293040793707509307051451399	false	-463989221200	-4459260420841	searchContents	2120	14:25:37
53519525997103898027244590	false	-463989221200	-4459260420841	searchContents	3260	14:25:37
403294415355463988513380138	false	-463989221200	-4459260420841	searchContents	3716	14:25:37
315882321506537520674337603	false	-463989221200	-4459260420841	searchContents	2063	14:25:37
622751063903109517827285424	false	-463989221200	-4459260420841	searchContents	4286	14:25:37
445926042084104773912099679	false	-463989221200	-4459260420841	searchContents	4115	14:25:37
425918795089979311707829186	false	-463989221200	-4459260420841	searchContents	2747	14:25:37
271035018066303680603318042	false	-463989221200	-4459260420841	searchContents	3944	14:25:37
26064583701090935598185484	false	-463989221200	-4459260420841	searchContents	3545	14:25:37
358215842856135232098737	false	-463989221200	-4459260420841	searchContents	2804	14:25:37
37993771617013239305865351	false	-463989221200	-4459260420841	searchContents	3773	14:25:37
57153303632492081240066392	false	-463989221200	-4459260420841	searchContents	3602	14:25:37

図 2 検証項目のスナップショット

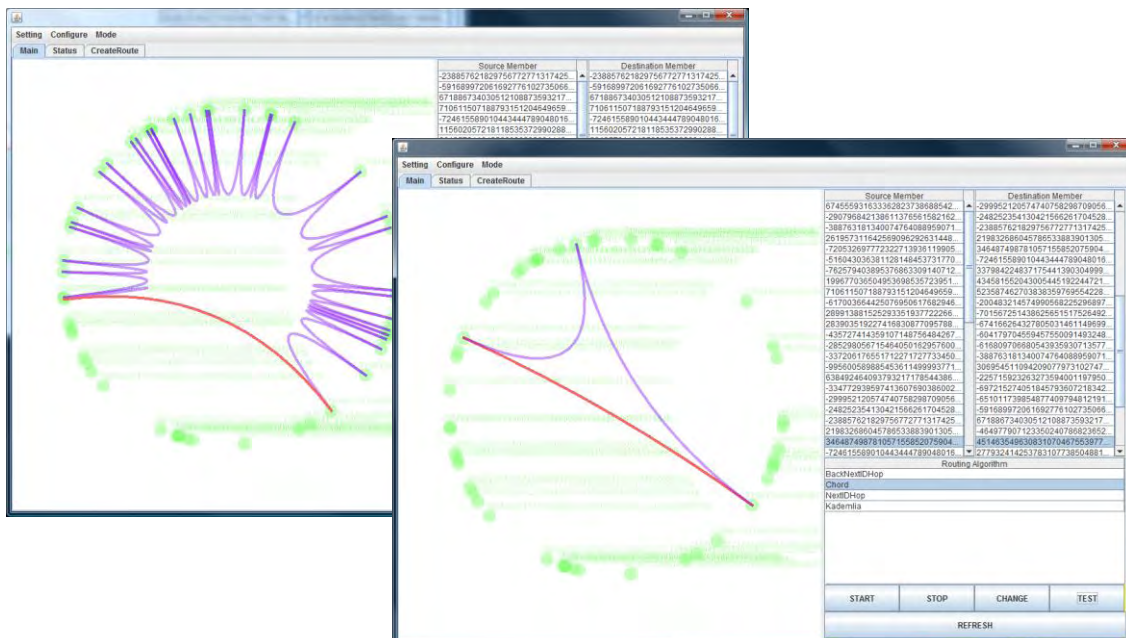


図 3 可視化のスナップショット

#### 4. 従来の技術(または機能)との相違

顕著な相違は Overlay Network の動作検証の部分にある。従来、提案されてきた Overlay Weaver や peeremu などの Overlay Network の検証機構では、単一の Overlay Network しか一度に検証する事ができず、検証のたびに Overlay Network を再構築する必要があり、検証するためのコストが大きかった。このコストは、実環境上で顕著にあらわれ、世界中に広域に広がるノードに対して再構築を行うには、オペレーション作業が作業効率上極めて大きなボトルネックになってくるので、Overlay Network の検証の敷居が非常に高くなる。本ミドルウェアを用いて検証基盤を構築した場合、Overlay Network の追加や変更手順が抽象化されているため、API を用いてルーティングアルゴリズムを記述するだけで良い。そのため、実環境であらかじめ検証基盤を構築しておけば、世界中の Overlay Network の研究者、分散システムの開発者が簡単に自身の提案する Overlay Network を検証する事が可能となる。

#### 5. 期待される効果

実環境での Overlay Network の検証が困難である事は 4 章で述べた。本ミドルウェアを Planet-lab 等の実インターネット上の実験環境に展開する事により、だれでも容易に Overlay Network の動作検証を行う事が可能となる。これにより Overlay Network の研究に拍車がかかる可能性がある。

#### 6. 普及(または活用)の見通し

Overlay Network のアルゴリズムコンテストを容易に行う事が可能と考えられる。本ミドルウェアを用いて Planet-lab に予め Overlay Network を構築しておき、様々なシチュエーション(特定のコンテンツを最速で取得 etc)に応じた Overlay Network をユーザが考え、反映し、検証結果が最も優れたアルゴリズム順に順位を付けるなどの競争が考えられる。なお、本ミドルウェアを用いて Planet-lab 上で 100 台の実機で検証が可能であることを現在確認しているため、今後 規模を拡大していき Overlay Network コンテストを開きたいと考えている。また検証の容易さから、個人や企業、教育機関に限らず Overlay Network に関わる人に使用していただきたいので、フリーソフトとして公開する予定である。

#### 7. 開発者名(所属)

水谷 后宏(奈良先端科学技術大学院大学 情報科学研究科 博士前期課程 2 年)

(参考)<http://inet-lab.naist.jp/~kimihiro-m/>