

ネットワークに潜在する依存関係のグラフ表現とその応用

2K-4

丸山 勝久 内藤 昭三

{maru,naito}@slab.ntt.co.jp

NTT ソフトウェア研究所

1 はじめに

コンピュータネットワークの発展により、利用者の要求するサービスは多様化および高度化し、そのようなサービスを実現するアプリケーションの構成要素（サービス構成要素）は複雑な関係を持つ。このような状況では、ネットワーク機器上に分散したサービス構成要素間の関係を、専門的な知識や経験を持つ管理者といえども完全に把握することは不可能である。サービスを健全あるいは効率的に利用および維持するためには、ネットワークの特性を系統的に解析し、その構造や機能を利用者の要求や環境に応じて適応的に変化させる仕組みが必要である。¹⁾

本論文では、ネットワークの構成（トポロジー）情報と、各機器上で実行するサービスの動作順序（シーケンス）情報に基づく依存関係に着目し、これらの依存関係を形式的に表現可能なネットワーク依存グラフ（NDG: Network Dependence Graph）を提案する。NDGを用いることで、指定サービスの実行に必要な最小構成要素の抽出や、複数の構成変更要求に対するサービス間の干渉要素の検出が可能である。よって、NDGはネットワーク構造や機能の適応的变化を実現する基盤となる。また、NDGは障害箇所の特定制無矛盾性を保証したサービスの合成にも応用でき、従来、管理者が行っていたネットワーク構成の設計、変更、運用の負担軽減が期待できる。

2 サービス間の依存関係解析

ネットワーク解析といった場合、パケットの到達経路などネットワーク構造に関する定性的解析と、トラフィックや遅延時間などの数値データに関する定量的解析がある。従来の解析手法の多くは、ネットワークの構造に基づくパケットの流れ（到達可能性やトラフィック）からネットワーク構造や状態を捉えていた。しかしながら、ネットワーク再構成を目的とする場合、ネットワーク上のサービスの機能変更あるいは機能保証を無視することはできず、単純にパケットの流れからサービスの機能を推測することは不可能である。ここで、サービスの機能とは、サービスの実行に関する入出力の関係を指す。

また、従来の解析手法では、ネットワーク機器間の経路に主に着目しているため、ホストやルータという粒度の大きなものを関連要素の基本単位としている。このため、このような手法に基づきネットワーク資源を再構成した場合、再構成による影響をサービスの粒度レベルで調整することができず、再構成がネットワークの効率的な利用につながらない恐れがある。

本論文では、ネットワーク機器間のパケットの流れではなく、サービス間の関連に着目し、ネットワークに潜

在する依存関係を解析する。その際、依存関係を持つ要素を、サービスを実現する断片という従来より小さい単位で扱う。これにより、ネットワーク構成変更が影響を及ぼす範囲（あるいは影響を及ぼさない範囲）をより正確に特定でき、再構成時にサービスの機能変更あるいは機能保証を実現可能である。

3 ネットワーク依存グラフ

ネットワーク依存グラフ（NDG）とは、プログラム依存グラフ（PDG）^{2), 3)}をネットワークに適用可能となるように定義したものである。ネットワーク機器の集合を H とする。各機器 $h \in H$ は、その機器上で提供しているサービスに関して、サービスの実体の集合 $Srv(h)$ 、サービスが参照するデータの集合 $Use(h)$ 、サービスが定義するデータの集合 $Def(h)$ を持つ。ここでは、これらの集合の要素 s をサービス節点と呼び、機器 h に属するサービス節点を (h, s) と表す。

NDG $G = (V, E)$ は、サービス節点の集合 V と各節点 $v \in V$ 間の矢印の集合 E からなる有向グラフである。矢印 $e \in E$ は、次に示す依存関係のどれか一つを表す。

データ依存関係: 機器 i 上のサービスが定義するデータ $d \in Def(i)$ の値が、機器 j 上のサービスが参照するデータ $u \in Use(j)$ の値に伝搬する場合、節点 $v = (i, d)$ から節点 $w = (j, u)$ にデータ依存がある。

制御依存関係: 機器 i 上のサービス $s \in Srv(i)$ の動作結果（正常あるいは異常）が、機器 j 上のサービスが参照するデータ $u \in Use(j)$ の値の有効性に影響を与える場合、節点 $v = (i, s)$ から節点 $w = (j, u)$ に制御依存がある。

サービス内依存関係: 機器 i 内において、サービス $s \in Srv(i)$ を実行し、データ $d \in Def(i)$ の値を定義するために必要な参照データ $u \in Use(i)$ が存在する場合、節点 $v = (i, u)$ から節点 $w1 = (i, s)$ および $w2 = (i, d)$ にサービス内依存関係がある。

データ依存関係と制御依存関係はネットワークのトポロジー（各種サービスの設定を含む）から、サービス内依存関係は機器上の各サービスに対する動作シーケンスから検出する。図1(a)(b)の情報を解析することで作成したネットワーク依存グラフを図1(c)に示す（図1(b)はftpのみに関する動作シーケンスである）。実線矢印はデータ依存関係、一点鎖線矢印は制御依存関係、点線矢印はサービス内依存関係を表す。また、入力矢印を持たない節点（uid, host, user, file）は、その節点が属する機器で設定されている固定情報（ファイルなど）を指す。

4 ネットワークスライシング

ネットワーク上のサービス断片間の依存関係がグラフで表現できると、従来、ソフトウェアの誤り検出や最適化に用いられていたプログラム依存関係解析に基づく手

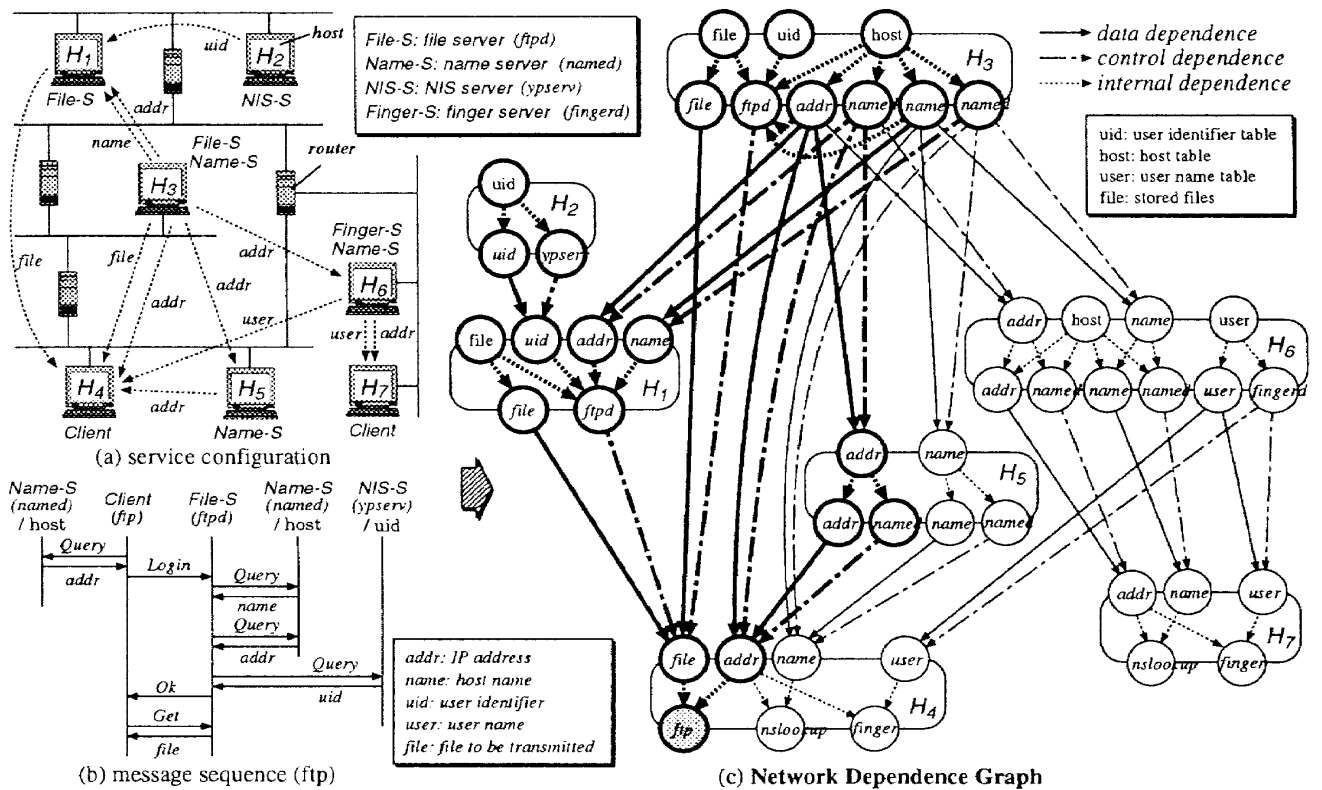


図 1: ネットワーク依存グラフ (NDG) およびネットワークスライス $S_b(H_4, \{ftp\})$

法をネットワーク再構成の手法として利用することが可能である。本論文では、NDGの応用の一つとして、ネットワークスライシングを定義し、スライシングによる障害箇所の特定期を示す。

ネットワークスライシングは、プログラムを分割するために考案されたプログラムスライシング⁴⁾をネットワークに適用したものである。ネットワーク機器の集合を H 、機器 $h \in H$ に属する節点を $V(h)$ をする。着目する機器 $h \in H$ と指定サービスの集合 $S \subseteq V(h)$ の組をスライシング基準 $C = \langle h, S \rangle$ とおく。ネットワークスライシング (Network Slicing) とは、着目する機器 h の指定サービス $s \in S$ が影響を与える関連サービス断片の集合、順方向スライス:

$$S_f(h, S) = \{(i, t) \in V \mid (h, s) \rightarrow^* (i, t) \wedge s \in S\}$$

あるいは、着目する機器 h の指定サービス $s \in S$ に影響を与える関連サービス断片の集合、

逆方向スライス:

$$S_b(h, S) = \{(i, t) \in V \mid (i, t) \rightarrow^* (h, s) \wedge s \in S\}$$

を、ネットワーク上のサービス全体から抽出することである。 \rightarrow^* は依存関係矢印の推移的閉包を指す(ここでは、3種類の依存関係を区別しない)。これらのスライスは、NDG $G = (V, E)$ において、節点 $v = (h, s)$ 、 $s \in S$ から依存関係矢印を順方向あるいは逆方向にたどることによって到達可能な節点を集めたものである。図1において、太線節点および矢印は機器 H_4 のサービス ftp に影響を与える逆方向スライス $S_b(H_4, \{ftp\})$ を表す。

いま、利用中のサービスに障害が発生した際、障害の発見箇所を基準として逆方向スライスを作成することで、

障害箇所を含む部分を切り出すことができる。さらに、障害が影響を与える範囲を順方向スライスにより見つけ、その範囲に含まれる個々のサービスを実際に行い、正常に動作しているサービスだけを取り除く、および、正常に動作していないサービスとの共通部分だけを抜き出すことで、障害箇所の特定期を絞り込むことが可能である。検出された障害箇所は、ネットワーク再構成における除去あるいは修復対象となる。

5 おわりに

現在、提案したネットワーク依存グラフの妥当性、たとえば、必要な依存関係を網羅しているかどうか、余分な依存関係を含まないか、などの検証を進めている。また、このグラフのさらなる応用例を考案中である。

謝辞 日頃御指導御討論いただく後藤厚宏リーダーはじめ、グループの皆様へ深く感謝します。

参考文献

- 1) 内藤昭三, 山本公洋, 丸山勝久, 藤浦豊徳: 適応的的自己組織化ネットワークシステムの提案, 情報処理学会 第55年全国大会, pp. 3-830-831 (1997).
- 2) Ferrante, J., Ottenstein, K. J. and Warren, J. D.: The Program Dependence Graph and Its Use in Optimization, *ACM TOPLAS*, Vol. 9, No. 3, pp. 319-349 (1987).
- 3) Jackson, D. and Rollins, E. J.: A New Model of Program Dependences for Reverse Engineering, *Proc. FSE*, pp. 2-10 (1994).
- 4) Weiser, M.: Program Slicing, *IEEE Trans. Softw. Eng.*, Vol. 10, No. 4, pp. 352-357 (1984).