无线网状网中自适应网络编码的MPTCP研究与设计

夏卓群[1,2] 陈志刚[1] 赵 明[1] 刘佳琦[1]

(中南大学信息科学与工程学院 长沙 410083)[1] (长沙理工大学计算机与通信工程学院 长沙 410076)[2]

摘 要 网络编码可以提高无线网状网的吞吐量,但是它在无线网状网中的实际应用是一个挑战性的问题。网络编码必须和TCP很好相容才能得到广泛应用。根据无线网状网的特点,研究和设计了一种自适应网络编码的多路径TCP。在提出的方案中,网络编码被加入到现有的网络系统,源节点根据目的节点编码数据块的自由度,调整发送编码的数据块,以降低多路径传输的数据报文的失序,提高网络的吞吐量,保证各个数据流之间的公平性。仿真结果表明设计的MPTCP有效。

关键词 无线网状网,网络编码,多路径TCP,自由度

Study and Design of Adaptive Network Coding MPTCP in Wireless Mesh Networks

XIA Zhuo-qun[1,2] CHEN Zhi-gang[1] ZHAO Ming[1] LIU Jia-qi[1]

(School of Information Science and Engineering, Central South University, Changsha 410083, China)[1]

(School of Computer and Telecommunication Engineering, Changsha University of Science and Technology, Changsha 410076 China)[2]

Abstract It is a challenge that network coding achieves wide use in wireless mesh networks. Network coding must be compatible or friendly with TCP in practice. We proposed a mechanism that naturally adds network coding to current network systems. In our scheme, according to destination feedback degree of freedom of encoding block, the source node modifies the coded data block and delays of packet delivery minimize reordering and timeouts. Using multipath routing, the source side can send more encode packets and the transmission rate becomes more faster, as a result, the network achieves higher throughput compared to TCP in the presence of lossy wireless links. Comprehensive simulations and results verify the validation of the theoretical results in the paper.

Keywords Wireless mesh networks, Network coding, Multipath TCP, Degree of freedom

无线网状网(Wireless mesh networks, WMNs)正成为社区和城市用户访问网络的重要选择[1]。网络编码(Network Coding)作为一种新型的数据编码方式,可以在网络中提高网络吞吐量、降低能耗等[2]。最近,越来越多的文献探讨将网络编码运用到无线网状网中,以提高网络的吞吐量[3-8]。虽然关于网络编码应用到无线网状网中的研究不少,但是在实际网络中应用网络编码的却很少。网络编码要在目前的网络上广泛使用,研究网络编码和现在网络上的主要传输控制协议(TCP)的相互融合,建立自适应网络编码的TCP就成为一个关键而棘手的问题。S. Katti等人[7]研究了网络编码如何影响TCP的吞吐量。但是,该文献没有研究TCP和网络编码的相互作用。Yong Huang等人[8]研究了TCP和网络编码的相互作用,并利用相互作用来提高TCP的吞吐量。然而,这篇文献缺少一个系统的分析模型,用以讨论网络编码对TCP吞吐量的影响。Jay Kumar Sundararajan等人[9]提出了以协议栈最少的变化将网络编码与TCP相互作用,但是没有针对无线网状网的特点,研究网络编码对无线网状网中TCP的影响。

无线网状网在一个发送端到一个接收端存在多条路径传输数据报文。这种多路径传输可以提高资源的利用率,而且与系统中采用单一路径相比,提高了鲁棒性且减少了性能的波动。因此,在无线网状网中采用多路径路由传输技术结合网络编码可以提高网络的吞吐量和公平性。本文考虑将网络编码和多路径TCP(MPTCP)结合,重点研究无线网状网中网络编码和MPTCP的相互影响,设计自适应网络编码的MPTCP。

由于多径路由技术采用并行方式传输数据报文,各路径的带宽、跳数以及节点处理能力的差异导致报文传输延时差别较大,在目的节点会出现报文乱序现象。当采用网络编码时,标准TCP的确认机制对收到的每个信息确认方法不再适应。其中最大的变化是,使用网络编码的目的节点只有收到足够的信息才开始解码得到原始信息。于是,目前网络中的ACK确认机制将被网络编码中反馈编码数据包的信息替代。本文采用新的确认机制,在多路径传输过程中最小化失序和超时问题。此外,当网络编码应用到无线网状网中时,由于从TCP层传递到IP层编码的数据报文受到传输容量的限制,

到稿日期:2009-10-13 返修日期:2009-12-20 本文受国家自然科学基金项目(60873082),总参预研基金(9140A15030308QT4801),中国博士后科学基金项目(20090451108),湖南省科技计划项目(2009RS3036),长沙市科技计划项目(K09ZD055-13)资助。

夏卓群(1977 -),男,博士生,讲师,主要研究方向为无线Mesh网络与网络编码,E-mail:xiazhuoqun@tom.com;**陈志刚**(1964-),男,博士,教授,博士生导师,CCF理事,主要研究方向为网络计算与分布式处理、计算机网络;**赵 明**(1975-),男,博士,主要研究方向为无线传感器网络。

网络吞吐量的提高受到影响。当采用多路径路由，网络可以提高 TCP 的性能。

本文主要研究无线网状网中网络编码和 MPTCP 的相互影响，将对协议栈进行最小的改变来建立自适应网络编码的 MPTCP；使用网络编码的反馈机制来解决多路径中的失序问题；且在多路径传输技术基础上增加 TCP 和 IP 层之间的编码层来提高网络的吞吐量。本文第 1 节研究和分析网络编码的反馈和无线网状网中的 MPTCP；第 2 节详细设计基于自适应网络编码的 MPTCP；第 3 节进行评估；最后对文章进行总结和展望。

1 网络编码和 MPTCP

1.1 网络编码

网络编码和传统的数据传输方式不同，它是一种融合编码和路由的信息交换技术，在传统存储转发的路由方法基础上，通过允许对多个数据包进行编码信息融合来增加单次传输的信息量，以提高网络整体性能。本文考虑在源节点和目的节点对信息进行随机线性编码。源节点将原始数据分成以块为单位的数据块，然后线性混合，编码后发送到网络。中间节点转发接收到的数据报文。当目的节点收到足够的编码数据报文时解码得到原始信息。

1.2 网络编码反馈

随机线性编码是基于数据块来进行编码和解码的，所以存在解码延迟。当在网络中应用网络编码时，许多应用需要发送一个持续的实时数据报文数据流。但网络编码需要把数据流分割成块，在一定时间内加以处理。目的节点如果只有在收到整个块以后才解码，那么吞吐量将有一个大的延迟。

在传输数据块时，数据的解码与整个数据块是否收到紧密相关，在实际应用中需要所有的数据报文已经恢复了数据才能解码。然而在数据流的应用中，解码先到达的数据报文将更早减少延迟。网络编码的性能不仅依赖于传输了多少数据，而且还依赖于传输的是哪一部分数据。解码过程中，对数据报文的延迟比编码数据整块的延迟影响更大。因此目的节点解码的开销不仅依赖于恢复的数据报文的数量，而且依赖于它们到达的顺序。

目的节点解码的数据报文受到窗口流动的数据报文的限制，在该窗口看到的数据报文有一个稳定的队列，所以编码的方案是即时在线的。所有看到的数据报文都是有序包含在发送端的队列。本文设定完美的反馈，在反馈中的丢失和延迟需要以后进一步研究。对于目的节点解码的延迟，如果目的节点接收到的一个数据报文是一个目的节点的自由度的头部数据时，它可以解码到其为止接收的所有数据报文。

在建立网络编码的反馈机制前，首先建立相关概念。本文所有的讨论对应一个源节点产生一个数据流。对每一个数据报文在一个有限的域 Γ_q，其大小为 q。k^{th} 个数据包对应源节点对应的标号为 k，标记为 p_k。

定义 1 看到数据包，[10]：一个节点看到数据包，如果它有足够的信息计算混合 (p_k+q)，$q=\sum_{l>k}\alpha_l p_l$，$\alpha_l\in\Gamma_q$，$l>k$。这里，q 是一个线形混合，包含了标号大于 k。

定义 2 节点的知识：是原始数据报文的所有线性混合的集合，这些集合基于它已收到的信息来计算。这些向量的系数形成一个向量的空间，定义为节点空间的知识。

定义 3 自由度：目的节点看到的数据报文数量。

网络编码在目的节点反馈自由度来确认可以解码的数据报文，不管是否已经接收到一个完整的数据报文。

命题 1 如果一个节点已经看到数据报文 p_k，根据已经得到的节点的知识，此节点精确地知道线性组合形成 (p_k+q)，其中 q 本身是一个只包含没有看到数据报文的线性组合。

根据命题 1，在目的节点解码得到编码的数据包时，自由度反映了已解码得到的原始数据的指标，而目前 TCP 主要是使用 ACK 确认最新到达的数据报文，这些数据报文按正确的顺序确保稳定的传输，同时反馈拥塞控制的信号。为了适应网络编码的使用，需要部分修改这个机制。关键的问题是在网络编码下的目的节点没有立即获得原始数据报文的信息，只有当足够的混合信息到达后，线性混合的数据报文才能解码得到原始的信息。因此，由 TCP 使用的按序列的数据报文的概念不能适应网络编码，而且，线性混合可能带来一些新的信息，对于目的节点即使它没有立即显示原始数据报文的信息。目前的 ACK 机制不允许目的节点在没有解码之前确认一个新的数据报文。

为适应网络编码，需要建立一种改进的 TCP 机制来确认收到的每个数据信息。每个到达的数据块信息都是通过自由度来反映的；一旦 n 个自由度被获得，n 个没有被解码的数据报文就可以被解码。本文研究一种实现 TCP 功能的新机制。基于确认收到的自由度，该机制可以实现可靠的传输和拥塞控制，不管目的节点是否立即显示了数据报文。解决的方法是在协议栈的传输控制层和网络层之间增加一个新的网络编码层。在增加编码层时，遵循以下两个条件：第一，源节点在拥塞窗口发送的总是随机线性混合的数据报文；第二，接收端确认的是自由度而不是原始数据报文。自由度反映了目的节点和源节点传输的数据报文的次序一致。新的 TCP 机制使用同样拥塞控制原则，配合拥塞窗口大小的原则与 TCP 一样。本文使用随机线性编码，实现端到端的 TCP，编码解码操作只在主机上执行。

1.3 MPTCP 与多路径传输模型

无线网状网自身有多个节点相互连接，在源节点和目的节点存在多条均能传输数据报文的路径，多路径技术比较适合无线网状网[6]，拓扑如图 1 所示。在多路径技术下，TCP 性能的研究也是目前研究的热点，有不少文献对多路径 TCP 的性能进行了探讨[11-13]，主要研究两方面问题。

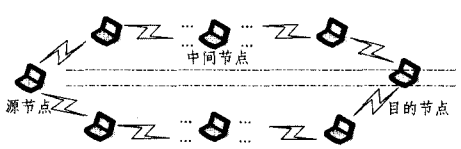

图 1 多路径传输拓扑

(1)失序和超时

第一，在多路径路由技术下，当一个 TCP 数据流经过瓶颈的链路时，由于同时有多条路径传输，将增加总的端到端带宽的有效性。但是由多条路径传输的各个数据报文到达目的节点的次序不相同，使得存在严重的数据报文失序。此外，多路径路由技术可能导致不同的平均往返时间(RTT)，而 RTT 不容易精确估计。在几条路径上的平均 RTT 可能比在长路径上的最大 RTT 时间短。因此在长路径上的发送端 TCP 可能过早地启动超时数据报文。而且，通过不同路径的数据报

文到达目的地会失序，从而启动双倍的 ACKs，这会触发不必要的 TCP 拥塞窗口调整，使其减少。

第二，数据报文可能会由于无线线路的错误而丢失，这种情况下，没有必要延迟其它数据报文。当一条路径被严重延迟或丢失时，被延迟的失序程序不能完全消除超时或多倍的双 ACKs。

(2)拥塞控制和公平性

在一对源节点和目的节点给定的多条可用路径中，设计可靠的拥塞控制算法，同时该算法也能有效利用多路径的容量是多路径传输技术 TCP 面临的另一问题。无线 TCP 中的性能包括检测无线的丢失，这样避免传递中不必要的延迟，得到拥塞的信号以及 TCP 的公平性。对每一条路径的拥塞控制要作出反应，而且要考虑无线网络中由于传输的丢失可能导致拥塞窗口的改变。

如果 TCP 流能公平竞争，则设计的拥塞控制具有友好性。TCP 公平性具有更高的要求。如果一个 TCP 协议没有更多的 TCP 流来取代 TCP 流本身，则该协议具有公平性，即 n 个 TCP 数据流共享一个瓶颈链路，每个 TCP 数据流会得到 $\frac{1}{n+1}$ 有效带宽。

本文改进文献[14]提出的数据报文调度模型，实现多路径路中使用背压式算法来调度数据流，保证多路径 TCP 的公平性。在模型中用 V 表示节点，用 $F\subseteq V^2$ 表示网络中的数据流，对每一个节点使用流入和流出。x_{ij}^f 表示数据报文的速率数据流 f 从节点 i 流到 j，y_f 表示数据流 f 新注入到源节点 $s(f)$ 的速率。$I_f(i)$，$O_f(i)\subseteq V$ 分别表示数据流 f 在节点 i 流入和流出的节点。在背压式方法中，如果节点 i 流入的数据流比总流出的数据流少，则认为 i 是稳定的，如式(1)所示。

$$\sum_{j\in O_{f(i)}} x_{ij}^f-\sum_{j\in I_{f(i)}} x_{ji}^f-\lambda y_f\geqslant 0 \tag{1}$$

如果 $s(f)=i$，则 $\lambda=1$，其它的为 0。式(1)表示平衡约束。同时有

$$x_{ij}^f\geqslant 0 \tag{2}$$

使用 $R=\{(r_{ij})\}$ 表示链路 (i,j) 可以由 MAC 层协议获得的平均速率。速率 R 由网络拓扑、信道条件、干扰等条件决定。本文假定有理想的 MAC 可以获得满足条件的速率。那么在 MAC 层的调度约束是

$$\left(\sum_{f\in F} x_{ij}^f\right)_{ij}\in R \tag{3}$$

式(1)－式(3)定义了网络支持的平均数据流速率集：$(y_f)_{f\in F}$ 数据集可以由网络 (V,F) 得到，如果存在速率 $(x_{ij}^f)_{i,j,f}$ 满足约束(1)－(3)。

$U_f(y_f)$ 表示根据数据流速率定义的数据函数。整个网络的效用函数是 $\sum_f U_f(y_f)$。整个效用函数最大化的方法是发现速率 $(y_f)_{f\in F}$，满足

$$\text{maxmize}\ \sum_f U_f(y_f)\ \text{subject to(1)－(3)} \tag{4}$$

上面的优化问题是一个凸函数，因为速率 R 是一个凸函数。它可以通过对偶公式使用梯度下降算法来求解，从而可以得到最佳的调度、路由、数据流控制算法。

数据流控制：数据流 f 在时间 t 的最佳速率是

$$y_f^*(t)=\underset{y_f>0}{\text{argmax}} U_f(y_f)-y_f q_{s(f)}^f(t) \tag{5}$$

$q_{s(f)}^f$ 为源节点为 $s(f)$ 的数据流 f 的队列长度，y_f 是 $q_{s(f)}^f$ 中含有多少个数据包的函数。

调度和路由：最佳的路由和调度可以由式(6)定义

$$r^*(t)=\underset{r\in R}{\text{argmax}}\sum_{i,j} r_{ij}\max_f(q_i^f(t)-q_j^f(t)) \tag{6}$$

式(6)定义的调度为背压式调度或最大权值调度。本文使用以上的调度方案确定无线 Mesh 网的多路径路由。为了充分利用多路径技术，确保与标准的 TCP 兼容，TCP 层不进行修改，数据报文调度层插入到 TCP 层和 IP 层。当 TCP 数据报文产生时，它首先被送到数据调度层，在这里根据一定的策略决定数据报文应该被采取的路由策略。路由的选择通过一个简单的接口和 IP 联系。然后 IP 层插入正确的值指示 ROUTE ID 的路由选择，转发数据报文到对应的路由。路由协议在每条发现的路由路径中寻找多条节点不相交的路径。所有这些有效的路径使用同时 TCP 传输数据报文，直到其中的路径都出现问题。当所有的路径都失败后，发现新的路径被启动。

2 自适应网络编码的 MPTCP 设计

根据在第 1 节研究的有关多路径路由和 TCP 的关系，以及建立的多路径传输模型，本节设计一个修改的多路径 TCP 来自适应无线网状网中的网络编码。这一节首先研究多路径 TCP 和网络编码的关系。然后设计适应网络编码的多路径 TCP 算法。

2.1 多路径 TCP 和网络编码

多路径 TCP 以前通过使用新到达的数据报文的确认机制来实现可靠传输。当网络编码应用到无线网状网中，确认的信息被修改了。在网络编码情况下，如果没有足够的信息到达，目的节点就不能获得原始数据报文的信息。于是，TCP 就不能马上得到数据报文的次序。目前的 ACK 确认机制没有允许接收端在没有解码之前确认一个数据报文。此外，TCP 希望在一定时间框架内按序接收数据报文，避免超时。由于使用多路径并行传输数据，路径的带宽、跳数和路径的处理能力的差别导致了任意的延迟和超时。目的节点将发生失序。本文在现有的 TCP 机制上进行最小的修改来确认每一个收到的信息。在 TCP 层与数据调度层中间增加一个编码层，如图 2 所示。目的节点反馈编码数据的自由度，最小化失序延迟和超时。

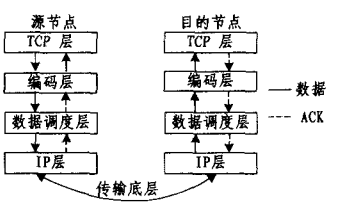

图 2 TCP 中的调度层和网络编码层

当每个来自 TCP 层的数据报文被线性编码发送到 IP 层时，由于信道的不可靠性、损耗，线性组合被送到网络的速率应该比 TCP 拥塞窗口处理的速率大。于是发送端通常发送一些冗余的信息到网络。然而，太多的冗余在单一的路径传输中容易受限，因为可以拥塞网络。使用多路径路由，发送端可以发送更多的编码数据报文，传输速率加快，这样对比在损耗的无线网络环境中的 TCP 网络能获得更高的吞吐量。

在使用多路径技术传输网络编码中，在目的节点一旦看到编码的数据报文就丢失源节点发送缓冲区的数据，也就是当一个数据报文在目的节点看到了就在源节点丢弃这个数据

报文。在解码这些数据后，目的节点能计算 q 获得 p_k。因此，即使目的节点没有解码 p_k，也不会出现信息因为丢弃而丢失。目的节点还未解码一个数据报文时，在看见的时候发出确认帧 ACK。目的节点使用一种确定性的编码方案来维护编码的数据报文，如果接收成功了，将同时导致目的节点知道它的下一个没有看到的数据报文。看到一个新的数据报文就能转化为一个新的自由度，从而可以获得较高的吞吐量。

2.2 自适应网络编码的多路径 TCP 实现

根据多路径 TCP 和网络编码之间的相互影响，将标准 MPTCP 作最小的改变来适应网络编码。本文采用在 TCP 层以下、数据调度层以上插入一层网络编码。详细描述如下。

(1)源节点

首先，采用随机线性码[15]作为网络编码方案。然后，编码的数据报文被发送到数据调度层，如果有多条路径可以被选择，一个数据报文的调度策略将决定选择哪条路径传输数据。调度策略将基于每条路径的反馈信息来决定选择路径。最后，编码的数据报文被送到链路层，通过多条路径进行传输。

(2)中间节点

一旦中间节点收到需要转发的数据报文，它们只要检验数据包的 ROUTE ID，确定数据报文的路由包含在它们的路由表中，转发数据报文到路由的下一跳。在目的节点收到 TCP 数据报文后，它产生一个确认帧。在确认数据报文里面包含了和接收到的 TCP 数据报文同样的 ROUTE ID。在中间节点按照相反的路径转发到发送端。

(3)目的节点

在目的节点，编码的数据报文被从 IP 层经数据调度层传递到编码层。当目的节点接收到数量足够的编码数据，就可以采用矩阵转换的方式恢复出完整原始报文。当目的节点的编码层收到少于完全解码的数据报文时，TCP 将发送编码的自由度来代替确认接收到的数据报文的 ACK。源节点将根据目的节点反馈的自由度调整发送的编码数据报文。

3 仿真评估

文章采用 NS 2.28 来对提出的自适应网络编码 MPTCP 进行仿真。拓扑结构如图 1 所示。两个 FTP 应用用于源节点和目的节点之间的通信。它们使用含网络编码的 TCP 和无网络编码的 TCP(TCP/NC)。其中一条链路带宽为 1.5 Mbps，另外一条链路带宽为 1Mbps，链路的缓存大小为 200。目的节点的 TCP 窗口设置为 100 个数据报文，数据报文大小为 1000 字节。根据 MPTCP 的主要功能，本文对丢包率和吞吐量、源节点的发送速度进行比较。

从图 3 可以得到在带有网络编码的网络中，丢包率比只采用多路径传输技术的要低。

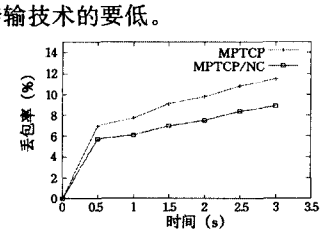

图 3 丢包率比较

在丢包率增加的情况下，图 4 表明 MPTCP/NC 的吞吐量没有大的改变，而 MPTCP 在丢包率 4%的情况下，吞吐量开始急剧下降。

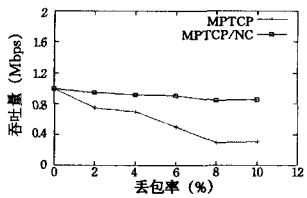

图 4 吞吐量比较

源节点的速率变化反映了拥塞的变化。从图 5 可以看到由于 MPTCP/NC 能较好地进行负载均衡，在链路上发生拥塞的机会少，因此源节点的速率变化不大，而 MPTCP 的速率变化大，应根据拥塞的情况不断调整发送速率。

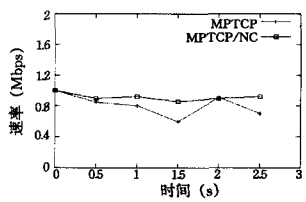

图 5 速率变化

结束语 本文研究了无线网状网中多路径 TCP 和网络编码之间的相互作用，并且设计了一种对协议栈很少改变的自适应网络编码的 MPTCP。为了解决失序和延迟的问题，本文采用确认机制里包含的自由度来反映可以解码的数量。

将来，我们通过优化无线网状网中的 MPTCP 来适应网络编码的实现，研究整合中间节点编码来提高网络的性能。

参 考 文 献

[1] Akyildiz I F, Wang X. A survey on wireless mesh networks[J]. IEEE Communications Magazine, 2005, 43(9)

[2] Ahlswede R, Cai N, Li S-Y R, et al. Network information flow [J]. IEEE Trans. Inf. Theory, 2000, 46(4): 1204-1216

[3] Alimi R, Li (Erran) Li, Ramjee R, et al. iPack: in-Network Packet Mixing for High Throughput Wireless Mesh Networks [C]// Proc. of IEEE INFOCOM '08. Anchorage, AK, March 2008

[4] Hamra A A, Barakat C, Turletti T. Network Coding forWireless Mesh Networks: A Case Study[C]// WOWMOM. 2006: 9-114

[5] Katti S, Katabi D, Balakrishnan H, et al. Symbol-level .Network Coding forWireless Mesh Networks[C]// SIGCOMM'08. Seattle, Washington, USA, August 2008

[6] Radunovicl B, Gkantsidisl C. Multipath Code Casting for Wireless Mesh Networks[R]. MSR-TR-2007-68. , March 2007

[7] Katti S, Rahul H, Hu W, et al. XORs in the air: Practical wireless network coding[C]// Proc. ACM SIGCOMM. Pisa, Italy, Sept. 2006

[8] Huang Yong, Ghaderi M, Towsley D, et al. TCP Performance in Coded Wireless Mesh Networks[C]// SECON '08. 5th Annual IEEE Communications Society Conference. June 2008: 179-187

[9] Sundararajan J K, Shah D, M'edard M, et al. Network coding meets TCP[C]// Proc. INFOCOM'09. Rio de Janeiro, Brazil, Apr. 2009

[10] Sundararajan J K, Shah D, M'edard M. ARQ for network coding [C]// Proc. of 2008 IEEE International Symposium on Information Theory (ISIT 2008)

(下转第 124 页)

联一个对应的 Condition/Action 标签，其中 Condition 表示执行该转换的条件，而 Action 表示该转换所执行的活动。

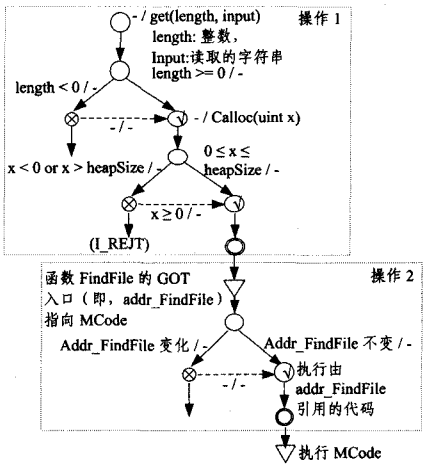

图 6 CFileFind::FindFile()堆溢出脆弱性

在图 6 中，操作 1(覆写 addr_FindFile)是操作 2(执行 MCode)的前置条件，通过上面的传播通道表示，而下面的传播通道(表示为 MCode)则是进入其他恶意操作的前置条件。在程序抽象模型和安全属性断言的基础上，通过调用 SPIN 模型检测器分析脆弱性，可以准确地揭示 MFC 库 CFileFind::FindFile()堆溢出脆弱性的安全违犯情况，并且获得触发脆弱性的执行轨迹，便于准确地定位和分析脆弱性。

结束语 本文提出了一种基于模型检测的二进制程序脆弱性分析框架，设计并实现了相应的脆弱性分析工具原型 MBVA。通过深入地研究二进制程序脆弱性的特点，提出了二进制程序的抽象模型以及基于有限状态自动机的软件脆弱性形式化表示方法。将软件脆弱性建模为一系列基本 FSM (eFSM)的集合。

根据软件脆弱性的形式化模型，为了便于自动化地检测和分析程序中的脆弱性，我们提出了基于事件系统的安全属性表示方法，用于指定推导出的安全断言。在此基础上，提出了基于模型检测的二进制程序脆弱性分析过程和算法，用于验证指定的程序模型是否满足一个安全属性的集合。本算法基于符号模型检测方法提供程序模型的可疑状态分析。在二进制程序脆弱性分析框架的基础上，设计并实现了 MBVA 工具原型，并且通过实验分析 Microsoft MFC 库 CFileFind::FindFile()堆溢出脆弱性，验证了此分析框架的有效性。

然而，对于一个给定的安全属性，正确地构造其形式化声明通常较为复杂。在下一步的工作中，将针对二进制程序模型，研究不同类型安全属性的自动构造方法。另外，对于通用的安全属性和脆弱性模式，如何提供形式化表示并且以自动方式构建其对应关系，也是需要进一步研究的问题。

参 考 文 献

[1] Singh P, Lakhotia A. Static verification of worm and virus behavior in binary executables using model checking[A]// 4th IEEE Information Assurance Workshop[C]. June 2003

[2] Sheyner O, Haines J, Jha S, et al. Automated generation and analysis of attack graphs[A]// Proc. 2002 IEEE Symposium on Security and Privacy[C]. 2002:254-265

[3] Ghosh M C, Simple A. State-based approaches to program-based anomaly detection[J]. ACM Transactions on Information and System Security, 2002, 5(3):203-237

[4] Clarke E, Grumberg O, Long D. Model Checking [M]. MIT Press, 1999

[5] Huth M, Ryan M. Logic in Computer Science, Modelling and Reasoning About Systems[M]. Second Edition. Cambridge University Press, 2004

[6] Ramakrishnan C, Sekar R. Model-based analysis of configuration vulnerabilities[J]. Journal of Computer Security, 2002, 10(1/2):189-209

[7] Chen H, Wagner D. MOPS: an Infrastructure for Examining Security Properties of Software[A]// Proceedings of the ACM Computer and Communications Security (CCS) Conference[C]. November 2002:235-244

[8] Ritchey R, Ammann P. Using model checking to analyze network vulnerabilities[A]// Proceedings of IEEE Symposium on Security and Privacy[C]. May 2000:156-165

[9] McCullough D. Specifications for Multi-level Security and a Hook-Up Property[A]// Poceedings of the 1987 IEEE Symposium on Research in Security and Privacy[C]. IEEE Press, May 1987

[10] Johnson D M, Thayer F J. Security and the Composition of Machines[A]// Proceedings of the Security Foundations Workshop [C]. Franconia, NH, June 1988:72-89

[11] Holzmann G J. The model checker SPIN[J]. IEEE Transactions on Software Engineering, 1997, 23(5):279-295

[12] The SMV System homepage[EB/OL]. URL. http://www.cs.cmu.edu/~modelcheck/smv.html

[13] The LaRS homepage[EB/OL]. URL: http://eis.jpl.nasa.gov/lars/

[14] Microsoft CFileFind:: FindFile Heap Overflow Vulnerability [EB/OL]. URL. http://cve.metre.org/cgi-bin/cvename.cgi?name=CVE-2007-4916

(上接第 109 页)

[11] Han Huaizhong, Shakkottai S, Hollot C V, et al. Multi-Path TCP: A Joint Congestion Control and Routing Scheme to Exploit Path Diversity in the Internet[J]. IEEE/ACM Transactions on Networking, 2006, 14(6):1260-1270

[12] Melakessou F, Sorger U, Suchanecki Z. MPTCP: Concept of a Flow Control Protocol Based on Multiple Paths for the Next Generation Internet[C]// Proceedings of the 7th International Symposium on Communications and Information Technologies ISCIT'07. Crowne Plaza Hotel, Darling Harbour, Sydney, Australia, October 16

[13] Chen Jiwei, Xu Kaixin, Gerla M. Multipath TCP in Lossy Wireless Environment[C]// IFIP Third Annual Mediterranean Ad Hoc Networking Workshop, (Med-Hoc-Net 2004). June 2004

[14] Radunovi'c B, Gkantsidis C, Gunawardena D. Peter Key Horizon: Balancing TCP over Multiple Paths in Wireless Mesh Network[C]// MobiCom '08: Proceedings of the 14th ACM international conference on mobile computing and networking. 2008: 247-258

[15] Ho T, Karger D R, Medard M, et al. The benefits of coding over routing in a randomized setting[C]// Hideki I. The 2003 IEEE International Symposium on Information Theory. San Jose: IEEE Press, 2003:442-447