Universität Trier

Fachbereich IV / Informatik

Prof. Dr. Peter Sturm
D-54286 Trier
Telefon 0651/201-3313
Fax 0651/201-3842
Email sturm@uni-trier.de

Philipp Müller
Email s4pimuel@uni-trier.de

Rechnernetze, Übungsblatt 6, Sommer 2024

Aufgabe 1: Konzepte

Erklären Sie in eigenen Worten, was Sliding-Window, TCP Tahoe, TCP Reno und TCP Vegas sind, wie sie funktionieren, und wofür man diese Techniken braucht. Schreiben Sie ihre Beschreibung in eine Textdatei oder PDF und zeichnen Sie mindestens eine Grafik, die irgendeines dieser Konzepte verdeutlicht. Schreiben Sie außerdem eine Liste aller Protokolle, die in der Vorlesung vorgekommen sind, und in welcher Schicht des ISO/OSI-Modells diese sich befinden (mit kurzer Begründung).

Aufgabe 2: DHCP

Fangen Sie sich mit Wireshark mindestens 4 DHCP-Pakete ein. Nutzen Sie ein Capture-Filter, in der PCAP-Datei sollten sich auschließlich DHCP-Pakete befinden. Überlegen Sie sich, was sie tun müssen, damit diese Pakete überhaupt gesendet/empfangen werden. Untersuchen Sie die Pakete. Machen Sie Screenshots der „Packet Details“ (der Bereich links unten) und beschreiben Sie in eigenen Worten, was Sie anhand der angezeigten Informationen über das Protokoll sagen können. Schreiben sie ihre Entdeckungen in eine Textdatei oder PDF.

Aufgabe 3: Nmap

Finden Sie folgende Dinge mithilfe von Nmap heraus. Falls Sie Windows verwenden, werden Sie eventuell zusätzlich dieses Tool benötigen, das Ihnen Informationen über IP-Adressen liefern kann. Verwenden Sie es mit „whois <ip-adresse>“. Beschreiben Sie die einzelnen Nmap-Befehle und deren Befehlselemente (-v, -T4, etc.), die sie verwendet haben. Sehen Sie sich dazu auch in Wireshark die Pakete an, die dabei gesendet werden.

a) Wie viele Hosts befinden sich in ihrem lokalen Klasse-C-Netz?

b) Welches Betriebssystem wird von scanme.nmap.org verwendet?

c) An welchem Datum wurde die Webseite nmap.org registriert?

d) Wie kann man möglichst effektiv eine größere Menge an Adressen nach offenen TCP-Ports scannen?

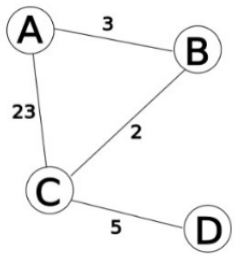
e) Wie funktioniert der SYN-Scan und für was kann man ihn verwenden?

f) Welches sind die offenen Ports, die bei Ihren bisherigen Nmap-Scans am häufigsten auftreten, und wofür werden sie verwendet?

Aufgabe 4: Routing

Informieren Sie sich über *dynamische*, und insbesondere *distanzvektorbasierte* Routingverfahren. Die Kosten zur Datenvermittlung zu anderen Rechnern werden in einem (Distanz-)Vektor in einer Tabelle gespeichert, die man Routingtabelle nennt.

In den Graphen werden Router als Knoten, Kosten als Kanten dargestellt. Erklären Sie mit Hilfe der Routingtabellen und dem Graphen des Netzes, was in jedem Schritt passiert, indem Sie einen Router raus suchen und die Veränderungen pro Runde analysieren und mit den Tabellen der anderen Router vergleichen. Entwickeln Sie darauf aufbauend einen Algorithmus (in Worten), der die Veränderungen in den Routingtabellen vollzieht. Was sind die Unterschiede zu *Link-State*-Verfahren?



Initialisierung:

von A	via A	via B	via C	via D
zu A				
zu B		3		
zu C			23	
zu D				

von B	via A	via B	via C	via D
zu A	3			
zu B				
zu C			2	
zu D				

von C	via A	via B	via C	via D
zu A	23			
zu B		2		
zu C				
zu D				5

von D	via A	via B	via C	via D
zu A				
zu B				
zu C			5	
zu D				

Aktualisierung:

von A	via A	via B	via C	via D
zu A				
zu B		3	25	
zu C		5	23	
zu D			28	

von B	via A	via B	via C	via D
zu A	3		25	
zu B				
zu C	26		2	
zu D			7	

von C	via A	via B	via C	via D
zu A	23	5		
zu B	26	2		
zu C				
zu D				5

von D	via A	via B	via C	via D
zu A			28	
zu B			7	
zu C			5	
zu D				

Aktualisierung:

von A	via A	via B	via C	via D
zu A				
zu B		3	25	
zu C		5	23	
zu D		10	28	

von B	via A	via B	via C	via D
zu A	3		7	
zu B				
zu C	8		2	
zu D	31		7	

von C	via A	via B	via C	via D
zu A	23	5		33
zu B	26	2		12
zu C				
zu D	51	9		5

von D	via A	via B	via C	via D
zu A			10	
zu B			7	
zu C			5	
zu D				

Aktualisierung und endgültiges Ergebnis:

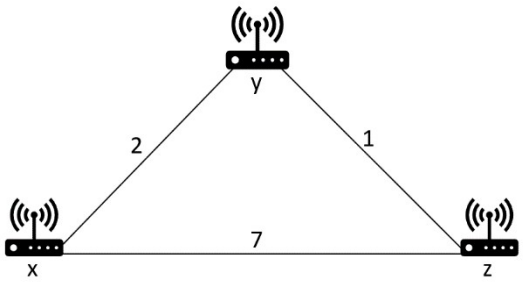
von A	via A	via B	via C	via D
zu A				
zu B		3	25	
zu C		5	23	
zu D		10	28	

von B	via A	via B	via C	via D
zu A	3		7	
zu B				
zu C	8		2	
zu D	13		7	

von C	via A	via B	via C	via D
zu A	23	5		15
zu B	26	2		12
zu C				
zu D	33	9		5

von D	via A	via B	via C	via D
zu A			10	
zu B			7	
zu C			5	
zu D				

a) Bestimmen Sie die Routingtabellen in jedem Schritt, indem Sie Ihren zuvor entwickelten Algorithmus anwenden. Geben Sie danach den kostengünstigsten Weg von Router „*z*“ zu Router „*x*“ an. Sie können die Lösungen einfach direkt in die Tabellen der PDF einfügen und diese in den Pull Request hinzufügen.



Von x	Via x	Via y	Via z
Zu x			
Zu y		2	
Zu z			7

Von y	Via x	Via y	Via z
Zu x	2		
Zu y			
Zu z			1

Von z	Via x	Via y	Via z
Zu x	7		
Zu y		1	
Zu z			

Von x	Via x	Via y	Via z
Zu x			
Zu y		2	8
Zu z		3	7

Von y	Via x	Via y	Via z
Zu x	2		8
Zu y			
Zu z	9		1

Von z	Via x	Via y	Via z
Zu x	4	3	
Zu y	9	1	
Zu z			

Von x	Via x	Via y	Via z
Zu x			
Zu y		2	4
Zu z		3	7

Von y	Via x	Via y	Via z
Zu x	2		4
Zu y			
Zu z	5		7

Von z	Via x	Via y	Via z
Zu x	7	3	
Zu y	5	7	
Zu z			

b) Die Kosten zwischen „*x*“ und „*y*“ steigen nun von 2 auf 7. Berechnen Sie die Routingtabellen mit Hilfe des Algorithmus. Ändert sich der kostengünstigste Pfad von „*z*“ nach „*x*“?

Von x	**Via x**	**Via y**	**Via z**
Zu x	//	//	//
Zu y	//	7	
Zu z	//		7

Von y	**Via x**	**Via y**	**Via z**
Zu x	7	//	
Zu y	//	//	//
Zu z		//	1

Von z	**Via x**	**Via y**	**Via z**
Zu x	7		//
Zu y		1	//
Zu z	//	//	//

Von x	**Via x**	**Via y**	**Via z**
Zu x	//	//	//
Zu y	//	7	8
Zu z	//	8	7

Von y	**Via x**	**Via y**	**Via z**
Zu x	7	//	8
Zu y	//	//	//
Zu z	14	//	1

Von z	**Via x**	**Via y**	**Via z**
Zu x	7	8	//
Zu y	14	1	//
Zu z	//	//	//

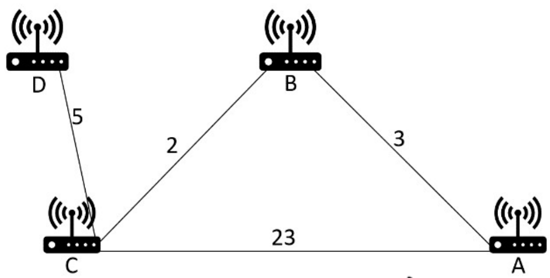
Von x	**Via x**	**Via y**	**Via z**
Zu x			
Zu y			
Zu z			

Von y	**Via x**	**Via y**	**Via z**
Zu x			
Zu y			
Zu z			

Von z	**Via x**	**Via y**	**Via z**
Zu x			
Zu y			
Zu z			

Ja, da die Abkürzung über y nichtmehr existiert

c) Sehen Sie sich den unteren Graphen an. Router „*D*“ fällt nun auf einmal aus. Beschreiben Sie, ob und wann die anderen Router merken, dass keine Verbindung mehr zu „*D*“ möglich ist.



Da kann der Count-to-Infinity bug passieren, da nur C mitbekommt, dass D nicht mehr funktioniert, aber noch mit A und B verbunden ist, die ja noch eine Verbindung zu D vorgeben, da sie nicht wissen das D nicht funktioniert. C routet also über A oder B, die dann über C routen. So entsteht ein Loop.

Abgabe

Die Abgabe in Form eines Pull-Requests in das Repository https://github.com/syssoft-ds/blackjack muß bis zum 17.7.2024 geschehen.