

컴퓨터 네트워킹: 패킷 스위칭, 회선 스위칭, 지연, 손실, 처리량

1.3.1 패킷 스위칭

네트워크 애플리케이션에서 엔드 시스템은 서로 메시지를 교환합니다. 메시지는 애플리케이션 설계자가 원하는 어떤 내용이라도 포함할 수 있습니다. 메시지는 제어 기능(예: 그림 1.2의 핸드셰이킹 예제에서 "Hi" 메시지)을 수행하거나, 이메일 메시지, JPEG 이미지, MP3 오디오 파일과 같은 데이터를 포함할 수 있습니다. 소스 엔드 시스템에서 목적지 엔드 시스템으로 메시지를 보내기 위해, 소스는 긴 메시지를 패킷이라는 더 작은 데이터 조각으로 나눕니다. 소스와 목적지 사이에서 각 패킷은 통신 링크와 패킷 스위치(주로 라우터와 링크 계층 스위치 두 가지 유형)를 통해 이동합니다. 패킷은 각 통신 링크에서 링크의 전체 전송 속도로 전송됩니다. 따라서 소스 엔드 시스템 또는 패킷 스위치가 L 비트의 패킷을 전송 속도 R 비트/초인 링크를 통해 보낼 때, 패킷 전송 시간은 L/R 초입니다.

스토어-앤-포워드 전송

대부분의 패킷 스위치는 링크 입력에서 스토어-앤-포워드 전송을 사용합니다. 스토어-앤-포워드 전송은 패킷 스위치가 패킷의 첫 번째 비트를 출력 링크로 전송하기 전에 전체 패킷을 수신해야 함을 의미합니다. 스토어-앤-포워드 전송을 더 자세히 살펴보기 위해, 그림 1.11과 같이 두 개의 엔드 시스템이 단일 라우터로 연결된 간단한 네트워크를 고려해 봅시다. 라우터는 일반적으로 여러 입력 링크를 가지며, 입력 패킷을 출력 링크로 스위칭하는 역할을 합니다. 이 간단한 예에서 라우터는 패킷을 하나의 입력 링크에서 유일한 출력 링크로 전달하는 단순한 작업을 수행합니다. 이 예에서 소스는 각각 L 비트로 구성된 세 개의 패킷을 목적지로 보내려고 합니다. 그림 1.11의 특정 시간 스냅샷에서 소스는 패킷 1의 일부를 전송했고, 패킷 1의 앞부분은 이미 라우터에 도착했습니다. 라우터가 스토어-앤-포워드 방식을 사용하기 때문에, 이 순간 라우터는 수신한 비트를 전송할 수 없으며, 대신 패킷의 비트를 버퍼에 저장(즉, "스토어")해야 합니다. 라우터는 패킷의 모든 비트를 수신한 후에야 출력 링크로 패킷을 전송(즉, "포워드")할 수 있습니다.

스토어-앤-포워드 전송에 대한 통찰을 얻기 위해, 소스가 패킷 전송을 시작한 시점부터 목적지가 전체 패킷을 수신할 때까지 걸리는 시간을 계산해 봅시다. (여기서는 전파 지연—비트가 거의 빛의 속도로 전선을 통해 이동하는 데 걸리는 시간—을 무시하며, 이는 1.4절에서 논의됩니다.) 소스는 시간 0에서 전송을 시작합니다. 시간 L/R 초에 소스는 전체 패킷을 전송했고, 라우터는 전체 패킷을 수신 및 저장합니다(전파 지연이 없으므로). 시간 L/R 초에 라우터는 전체 패킷을 수신했으므로 출력 링크를 통해 목적지로 패킷 전송을 시작할 수 있습니다. 시간 $2L/R$ 초에 라우터는 전체 패킷을 전송했고, 목적지는 전체 패킷을 수신합니다. 따라서 총 지연은 $2L/R$ 입니다. 만약 스위치가 전체 패킷을 수신하기 전에 도착하는 비트를 즉시 포워딩한다면, 총 지연은 L/R 이 될 것이며, 이는 라우터에서 비트가 지체되지 않기 때문입니다. 하지만 1.4절에서 논의하겠지만, 라우터는 포워딩 전에 전체 패킷을 수신, 저장, 처리해야 합니다.

이제 소스가 첫 번째 패킷 전송을 시작한 시점부터 목적지가 세 개의 패킷을 모두 수신할 때까지 걸리는 시간을 계산해 봅시다. 앞서와 같이, 시간 L/R 초에 라우터는 첫 번째 패킷을

포워딩하기 시작합니다. 또한 시간 L/R 초에 소스는 첫 번째 패킷 전송을 완료했으므로 두 번째 패킷 전송을 시작합니다. 따라서 시간 $2L/R$ 초에 목적지는 첫 번째 패킷을 수신하고, 라우터는 두 번째 패킷을 수신합니다. 마찬가지로, 시간 $3L/R$ 초에 목적지는 첫 번째와 두 번째 패킷을 수신하고, 라우터는 세 번째 패킷을 수신합니다. 마지막으로, 시간 $4L/R$ 초에 목적지는 세 개의 패킷을 모두 수신합니다!

이제 소스에서 목적지로 한 개의 패킷을 각 속도가 R 인 N 개의 링크로 구성된 경로를 통해 보내는 일반적인 경우를 고려해 봅시다(즉, 소스와 목적지 사이에 $N-1$ 개의 라우터가 있음). 위와 동일한 논리를 적용하면, 엔드-투-엔드 지연은 다음과 같습니다:

$$d_{\text{end-to-end}} = N * (L / R) \text{ [식 1.1]}$$

$d_{\text{end-to-end}}$: 엔드-투-엔드 지연 (단위: 초)

- 소스 엔드 시스템에서 패킷 전송이 시작된 시점부터 목적지 엔드 시스템이 해당 패킷을 완전히 수신할 때까지 걸리는 총 시간입니다. 이 지연은 경로상의 모든 노드(라우터 및 호스트)에서 발생하는 지연의 합으로 구성됩니다.

N : 경로상의 링크 수

- 소스와 목적지 사이에 있는 통신 링크의 총 개수입니다. 이는 소스와 목적지 사이에 있는 라우터의 개수에 1을 더한 값과 같습니다(즉, $N = (\text{라우터 수} + 1)$). 예를 들어, 소스와 목적지 사이에 라우터가 하나 있다면, 링크 수는 2입니다.

L : 패킷의 길이 (단위: 비트)

- 전송되는 패킷의 크기, 즉 패킷에 포함된 비트 수입니다. 예를 들어, 패킷이 1,000비트로 구성되어 있다면, $L = 1,000$ 비트입니다.

R : 링크의 전송 속도 (단위: 비트/초, bps)

- 각 링크를 통해 비트가 전송되는 속도입니다. 예를 들어, 10Mbps 이더넷 링크의 경우, $R = 10,000,000$ 비트/초입니다. 이 식에서는 모든 링크가 동일한 전송 속도 R 를 가진다고 가정합니다.

이제 P 개의 패킷을 N 개의 링크로 보내는 경우의 지연을 계산해 보는 것을 시도해 볼 수 있습니다.

큐잉 지연과 패킷 손실

각 패킷 스위치에는 여러 개의 링크가 연결되어 있습니다. 각 연결된 링크에 대해 패킷 스위치는 라우터가 해당 링크로 보내기 위해 준비한 패킷을 저장하는 출력 버퍼(또는 출력 큐)를 가지고 있습니다. 출력 버퍼는 패킷 스위칭에서 핵심적인 역할을 합니다. 도착한 패킷이 링크로 전송되어야 하지만, 해당 링크가 다른 패킷의 전송으로 인해 사용 중이라면, 도착한 패킷은 출력 버퍼에서 대기해야 합니다. 따라서 스토어-앤-포워드 지연 외에도 패킷은 출력 버퍼 큐잉 지연을 겪습니다. 이러한 지연은 네트워크의 혼잡 수준에 따라 달라지며 가변적입니다. 버퍼 공간의 크기는 유한하기 때문에, 도착한 패킷이 다른 패킷들로 이미 가득 찬 버퍼를 만날 수 있습니다. 이 경우 패킷 손실이 발생합니다—도착한 패킷 또는 이미 대기 중인 패킷 중 하나가 버려집니다.

그림 1.12는 간단한 패킷 스위칭 네트워크를 보여줍니다. 그림 1.11과 마찬가지로, 패킷은 3차원 슬래브로 표현됩니다. 슬래브의 너비는 패킷의 비트 수를 나타냅니다. 이 그림에서는 모든 패킷이 동일한 너비, 즉 동일한 길이를 가집니다. 호스트 A와 B가 호스트 E로 패킷을 보내고 있다고 가정해 봅시다. 호스트 A와 B는 먼저 100Mbps 이더넷 링크를 통해 첫 번째 라우터로 패킷을 보냅니다. 라우터는 이 패킷들을 15Mbps 링크로 전달합니다. 짧은 시간 동안 라우터에 도착하는 패킷의 도착 속도(비트/초로 변환했을 때)가 15Mbps를 초과하면, 패킷이 링크로 전송되기 전에 링크의 출력 버퍼에 대기하면서 라우터에서 혼잡이 발생합니다. 예를 들어, 호스트 A와 B가 동시에 연속적으로 5개의 패킷을 버스트로 보낸다면, 이 패킷들 중 대부분은 큐에서 대기하는 시간을 겪게 됩니다. 이 상황은 일상적인 상황, 예를 들어 은행 창구에서 줄을 서거나 톨게이트 앞에서 대기하는 것과 완전히 유사합니다. 이 큐잉 지연은 1.4절에서 더 자세히 살펴볼 것입니다.

포워딩 테이블과 라우팅 프로토콜

앞서 우리는 라우터가 연결된 통신 링크 중 하나로 도착한 패킷을 받아 다른 연결된 통신 링크로 전달한다고 설명했습니다. 하지만 라우터는 패킷을 어느 링크로 전달할지 어떻게 결정할까요? 패킷 포워딩은 컴퓨터 네트워크의 종류에 따라 다양한 방식으로 이루어집니다. 여기서는 인터넷에서 이 과정이 어떻게 이루어지는지 간략히 설명하겠습니다.

인터넷에서 모든 엔드 시스템은 IP 주소라는 주소를 가집니다. 소스 엔드 시스템이 목적지 엔드 시스템으로 패킷을 보내려고 할 때, 소스는 패킷의 헤더에 목적지의 IP 주소를 포함시킵니다. 우편 주소와 마찬가지로, 이 주소는 계층적 구조를 가집니다. 패킷이 네트워크의 라우터에 도착하면, 라우터는 패킷의 목적지 주소 일부를 검사하여 패킷을 인접한 라우터로 전달합니다. 더 구체적으로, 각 라우터는 목적지 주소(또는 목적지 주소의 일부)를 라우터의 출력 링크에 매핑하는 포워딩 테이블을 가지고 있습니다. 패킷이 라우터에 도착하면, 라우터는 주소를 검사하고 이 목적지 주소를 사용하여 포워딩 테이블을 검색하여 적절한 출력 링크를 찾습니다. 그런 다음 라우터는 패킷을 이 출력 링크로 전달합니다.

엔드-투-엔드 라우팅 과정은 지도를 사용하지 않고 길을 물으며 운전하는 운전자와 유사합니다. 예를 들어, 조(Joe)가 필라델피아에서 플로리다 올랜도의 156 레이크사이드 드라이브로 운전한다고 가정해 봅시다. 조는 먼저 동네 주유소로 가서 올랜도의 156 레이크사이드 드라이브로 가는 길을 묻습니다. 주유소 직원은 주소의 플로리다 부분을 확인하고 조에게 주유소 바로 옆에 있는 I-95 남쪽 방향 고속도로 입구로 가라고 알려줍니다. 또한 플로리다에 도착하면 다른 사람에게 다시 물어보라고 조언합니다. 조는 I-95 남쪽으로 운전하여 플로리다 잭슨빌에 도착하고, 그곳에서 또 다른 주유소 직원에게 길을 묻습니다. 직원은 주소의 올랜도 부분을 확인하고 조에게 I-95를 따라 데이тона 비치까지 가서 다시 물어보라고 알려줍니다. 데이тона 비치에서 또 다른 주유소 직원이 주소의 올랜도 부분을 확인하고 조에게 I-4를 타고 바로 올랜도로 가라고 알려줍니다. 조는 I-4를 타고 올랜도 출구에서 내립니다. 조는 또 다른 주유소 직원에게 가서 이번에는 주소의 레이크사이드 드라이브 부분을 확인받고, 레이크사이드 드라이브로 가는 길을 안내받습니다. 레이크사이드 드라이브에 도착한 조는 자전거를 탄 아이에게 최종 목적지를 묻고, 아이는 주소의 156 부분을 확인하여 집을 가리킵니다. 마침내 조는 최종 목적지에 도착합니다. 이 비유에서 주유소 직원과 자전거를 탄 아이들은 라우터에 해당합니다.

우리는 라우터가 패킷의 목적지 주소를 사용하여 포워딩 테이블을 검색하고 적절한 출력 링크를 결정한다는 것을 배웠습니다. 하지만 이는 또 다른 질문을 낳습니다: 포워딩

테이블은 어떻게 설정되는가? 각 라우터에서 수동으로 설정되는가, 아니면 인터넷은 더 자동화된 절차를 사용하는가? 이 문제는 5장에서 깊이 연구할 것입니다. 하지만 여기서 호기심을 자극하기 위해, 인터넷은 포워딩 테이블을 자동으로 설정하는 여러 특별한 라우팅 프로토콜을 사용한다고 언급하겠습니다. 예를 들어, 라우팅 프로토콜은 각 라우터에서 각 목적지까지의 최단 경로를 결정하고, 최단 경로 결과를 사용하여 라우터의 포워딩 테이블을 구성할 수 있습니다.

1.3.2 회선 스위칭

링크와 스위치로 구성된 네트워크를 통해 데이터를 이동시키는 두 가지 근본적인 접근 방식은 회선 스위칭과 패킷 스위칭입니다. 이전 하위 섹션에서 패킷 스위칭 네트워크를 다루었으므로, 이제 회선 스위칭 네트워크에 대해 살펴보겠습니다.

회선 스위칭 네트워크에서는 엔드 시스템 간의 통신 세션 동안 필요한 경로 상의 자원(버퍼, 링크 전송 속도)이 해당 세션 기간 동안 예약됩니다. 반면, 패킷 스위칭 네트워크에서는 이러한 자원이 예약되지 않으며, 세션의 메시지는 필요에 따라 자원을 사용합니다. 이로 인해 통신 링크에 접근하기 위해 대기(즉, 큐잉)해야 할 수 있습니다. 간단한 비유로, 두 개의 레스토랑을 생각해 봅시다. 하나는 예약이 필요한 레스토랑이고, 다른 하나는 예약을 요구하지도, 받지도 않는 레스토랑입니다. 예약이 필요한 레스토랑의 경우, 집을 나서기 전에 전화로 예약해야 하는 번거로움이 있습니다. 하지만 레스토랑에 도착하면 원칙적으로 즉시 자리에 앉아 음식을 주문할 수 있습니다. 반면, 예약이 필요 없는 레스토랑에서는 예약을 할 필요가 없지만, 도착했을 때 자리가 날 때까지 기다려야 할 수 있습니다.

전통적인 전화 네트워크는 회선 스위칭 네트워크의 대표적인 예입니다. 한 사람이 전화 네트워크를 통해 다른 사람에게 정보(음성 또는 팩스)를 보내려고 할 때 어떤 일이 일어나는지 생각해 봅시다. 발신자가 정보를 보내기 전에 네트워크는 발신자와 수신자 간의 연결을 설정해야 합니다. 이는 발신자와 수신자 간의 경로에 있는 스위치들이 해당 연결에 대한 연결 상태를 유지하는 진정한 연결입니다. 전화 용어로는 이 연결을 ****회선(circuit)****이라고 부릅니다. 네트워크가 회선을 설정할 때, 연결 기간 동안 네트워크의 링크에서 일정한 전송 속도(각 링크의 전송 용량의 일부)를 예약합니다. 이 발신자-수신자 연결을 위해 일정한 전송 속도가 예약되었기 때문에, 발신자는 보장된 일정한 속도로 데이터를 수신자에게 전송할 수 있습니다.

그림 1.13은 회선 스위칭 네트워크를 보여줍니다. 이 네트워크에서는 네 개의 회선 스위치가 네 개의 링크로 상호 연결되어 있습니다. 각 링크는 네 개의 회선을 가지므로, 각 링크는 최대 네 개의 동시 연결을 지원할 수 있습니다. 호스트(예: PC, 워크스테이션)는 각각 하나의 스위치에 직접 연결되어 있습니다. 두 호스트가 통신하려고 할 때, 네트워크는 두 호스트 간의 전용 엔드-투-엔드 연결을 설정합니다. 따라서 호스트 A가 호스트 B와 통신하려면, 네트워크는 두 개의 링크 각각에서 하나의 회선을 예약해야 합니다. 이 예에서, 전용 엔드-투-엔드 연결은 첫 번째 링크의 두 번째 회선과 두 번째 링크의 네 번째 회선을 사용합니다. 각 링크가 네 개의 회선을 가지므로, 엔드-투-엔드 연결이 사용하는 각 링크에 대해 연결은 링크의 전체 전송 용량의 1/4을 연결 기간 동안 할당받습니다. 예를 들어, 인접 스위치 간의 각 링크의 전송 속도가 1 Mbps라면, 각 엔드-투-엔드 회선 스위칭 연결은 250 kbps의 전용 전송 속도를 갖게 됩니다.

패킷 스위칭 네트워크에서의 데이터 전송

패킷 스위칭 네트워크(예: 인터넷)에서 한 호스트가 다른 호스트로 패킷을 보낼 때 어떤 일이 일어나는지 살펴보겠습니다. 회선 스위칭과 마찬가지로 패킷은 일련의 통신 링크를 통해 전송됩니다. 하지만 회선 스위칭과 달리, 패킷은 링크 자원을 전혀 예약하지 않고 네트워크로 전송됩니다. 만약 다른 패킷들이 동시에 같은 링크를 통해 전송되어야 해서 링크가 혼잡하다면, 패킷은 전송 링크의 발신 측에서 버퍼에 대기해야 하며, 이로 인해 지연이 발생합니다. 인터넷은 패킷을 적시에 전달하기 위해 최선을 다하지만, 어떠한 보장도 하지 않습니다.

회선 스위칭 네트워크에서의 다중화

회선 스위칭 네트워크에서 링크의 회선은 주파수 분할 다중화(FDM) 또는 **시간 분할 다중화(TDM)**를 통해 구현됩니다.¹

- **FDM:** 링크의 주파수 스펙트럼을 링크를 통해 설정된 연결들 사이에 나눕니다. 구체적으로, 링크는 연결 기간 동안 각 연결에 특정 주파수 대역을 할당합니다. 전화 네트워크에서는 이 주파수 대역의 폭이 일반적으로 4kHz(초당 4,000 사이클)입니다. 이 대역의 폭은 대역폭이라고 불립니다. FM 라디오 방송국도 88MHz에서 108MHz 사이의 주파수 스펙트럼을 공유하기 위해 FDM을 사용하며, 각 방송국은 특정 주파수 대역을 할당받습니다.
- **TDM:** 시간을 고정된 지속 시간의 프레임으로 나누고, 각 프레임은 고정된 수의 시간 슬롯으로 세분화됩니다. 네트워크가 링크를 통해 연결을 설정할 때, 모든 프레임에서 하나의 시간 슬롯을 해당 연결에 할당합니다. 이 슬롯은 해당 연결만을 위해 전용으로 사용되며, 각 프레임마다 하나의 슬롯이 연결의 데이터를 전송하는 데 사용됩니다.

그림 1.14는 최대 4개의 회선을 지원하는 특정 네트워크 링크에 대한 FDM과 TDM을 보여줍니다. FDM에서는 주파수 영역이 각각 4kHz 대역폭을 가진 4개의 대역으로 분할됩니다. TDM에서는 시간 영역이 프레임으로 분할되며, 각 프레임에는 4개의 시간 슬롯이 있습니다. 각 회선은 순환하는 TDM 프레임에서 동일한 전용 슬롯을 할당받습니다. TDM에서 회선의 전송 속도는 프레임 속도에 슬롯의 비트 수를 곱한 값입니다. 예를 들어, 링크가 초당 8,000 프레임을 전송하고 각 슬롯이 8비트로 구성되어 있다면, 각 회선의 전송 속도는 64kbps입니다.

패킷 스위칭 지지자들은 회선 스위칭이 침묵 기간 동안 전용 회선이 유휴 상태로 남아 자원을 낭비한다고 주장해 왔습니다. 예를 들어, 전화 통화에서 한 사람이 말을 멈추면, 연결 경로의 링크에 있는 유휴 네트워크 자원(주파수 대역 또는 시간 슬롯)은 다른 진행 중인 연결에서 사용할 수 없습니다. 또 다른 예로, 회선 스위칭 네트워크를 사용하여 원격으로 X-ray 이미지를 확인하는 방사선 전문의를 생각해 봅시다. 방사선 전문의는 연결을 설정하고 이미지를 요청한 후, 이미지를 검토하고 새로운 이미지를 요청합니다. 이 검토 기간 동안 네트워크 자원은 연결에 할당되어 있지만 사용되지 않아(즉, 낭비) 됩니다. 또한 패킷 스위칭 지지자들은 엔드-투-엔드 회선을 설정하고 엔드-투-엔드 전송 용량을 예약하는 것이 복잡하며, 경로 상의 스위치 동작을 조정하기 위해 복잡한 신호 소프트웨어가 필요하다고 지적합니다.

회선 스위칭에 대한 논의를 마무리하기 전에, 수치 예제를 통해 더 깊은 통찰을 얻어 봅시다. 호스트 A에서 호스트 B로 640,000 비트의 파일을 회선 스위칭 네트워크를 통해 보내는 데 걸리는 시간을 계산해 봅시다. 네트워크의 모든 링크가 24개의 슬롯을 가진 TDM을 사용하며, 비트 속도가 1.536Mbps라고 가정합니다. 또한 엔드-투-엔드 회선을 설정하는 데

500ms가 걸린다고 가정합니다. 파일을 보내는 데 걸리는 시간은 얼마일까요? 각 회선의 전송 속도는 $(1.536\text{Mbps})/24 = 64\text{kbps}$ 입니다. 따라서 파일 전송 시간은 $(640,000\text{비트})/(64\text{kbps}) = 10\text{초}$ 입니다. 여기에 회선 설정 시간을 더하면, 파일을 보내는 데 총 10.5초가 걸립니다. 전송 시간은 링크 수와 무관합니다. 즉, 엔드-투-엔드 회선이 한 개의 링크를 통과하든 백 개의 링크를 통과하든 전송 시간은 10초입니다. (실제 엔드-투-엔드 지연에는 전파 지연도 포함되며, 이는 1.4절에서 다룹니다.)

패킷 스위칭 대 회선 스위칭

회선 스위칭과 패킷 스위칭을 설명했으니, 이제 둘을 비교해 보겠습니다. 패킷 스위칭 비판자들은 패킷 스위칭이 가변적이고 예측 불가능한 엔드-투-엔드 지연(주로 가변적이고 예측 불가능한 큐잉 지연 때문) 때문에 실시간 서비스(예: 전화 통화, 화상 회의)에 적합하지 않다고 주장해 왔습니다. 반면, 패킷 스위칭 지지자들은 (1) 패킷 스위칭이 회선 스위칭보다 전송 용량을 더 잘 공유하며, (2) 회선 스위칭보다 더 간단하고 효율적이며 구현 비용이 적게 든다고 주장합니다. 패킷 스위칭과 회선 스위칭에 대한 흥미로운 논의는 [Molinero-Fernandez 2002]에서 확인할 수 있습니다. 일반적으로 레스토랑 예약의 번거로움을 싫어하는 사람들은 회선 스위칭보다 패킷 스위칭을 선호합니다.

패킷 스위칭이 왜 더 효율적일까요? 간단한 예제를 살펴보겠습니다. 사용자가 1Mbps 링크를 공유한다고 가정합니다. 또한 각 사용자는 활동 기간 동안 100kbps의 일정한 속도로 데이터를 생성하고, 비활성 기간 동안에는 데이터를 생성하지 않는다고 가정합니다. 사용자가 10%의 시간 동안만 활동적이라고 가정합니다(나머지 90%는 유휴 상태로 커피를 마시고 있음). 회선 스위칭에서는 각 사용자에게 항상 100kbps가 예약되어야 합니다. 예를 들어, 회선 스위칭 TDM에서 1초 프레임이 100ms의 10개 시간 슬롯으로 나뉘면, 각 사용자는 프레임당 하나의 시간 슬롯을 할당받습니다. 따라서 회선 스위칭 링크는 10명(= $1\text{Mbps}/100\text{kbps}$)의 동시 사용자를 지원할 수 있습니다.

패킷 스위칭에서는 특정 사용자가 활동 중일 확률이 0.1(즉, 10%)입니다. 35명의 사용자가 있을 때, 11명 이상이 동시에 활동할 확률은 약 0.0004입니다. (이 확률은 숙제 문제 P8에서 계산 방법이 설명됨.) 10명 이하의 사용자가 동시에 활동할 때(확률 0.9996), 데이터의 총 도착 속도는 링크의 출력 속도인 1Mbps 이하입니다. 따라서 10명 이하의 사용자가 활동 중일 때는 패킷이 회선 스위칭처럼 지연 없이 링크를 통과합니다. 10명을 초과하는 사용자가 동시에 활동하면, 패킷의 총 도착 속도가 링크의 출력 용량을 초과하므로 출력 큐가 증가하기 시작합니다. (총 입력 속도가 1Mbps 이하로 떨어질 때까지 큐는 계속 증가하며, 이후 큐 길이는 줄어듭니다.) 이 예에서 10명을 초과하는 사용자가 동시에 활동할 확률이 매우 작기 때문에, 패킷 스위칭은 회선 스위칭과 거의 동일한 성능을 제공하면서도 3배 이상의 사용자를 지원할 수 있습니다.

두 번째 간단한 예제를 살펴보겠습니다. 10명의 사용자가 있고, 한 사용자가 갑자기 1,000비트 패킷 1,000개를 생성하고 다른 사용자는 패킷을 생성하지 않는다고 가정합니다. TDM 회선 스위칭에서 프레임당 10개의 슬롯이 있고 각 슬롯이 1,000비트로 구성되어 있다면, 활동적인 사용자는 프레임당 하나의 시간 슬롯만 사용할 수 있으며, 나머지 9개의 슬롯은 유휴 상태로 남습니다. 활동적인 사용자의 100만 비트 데이터가 모두 전송되는 데 10초가 걸립니다. 반면, 패킷 스위칭에서는 다른 사용자가 패킷을 생성하지 않으므로 활동적인 사용자가 1Mbps의 전체 링크 속도로 패킷을 연속적으로 보낼 수 있습니다. 이 경우, 활동적인 사용자의 모든 데이터는 1초 이내에 전송됩니다.

위 예제들은 패킷 스위칭의 성능이 회선 스위칭보다 우수할 수 있는 두 가지 방식을 보여줍니다. 또한, 링크의 전송 속도를 여러 데이터 스트림 간에 공유하는 두 방식의 중요한 차이점을 강조합니다. 회선 스위칭은 수요와 관계없이 전송 링크의 사용을 미리 할당하며, 할당되었지만 사용되지 않는 링크 시간은 낭비됩니다. 반면, 패킷 스위칭은 수요에 따라 링크 사용을 할당합니다. 링크 전송 용량은 패킷 단위로, 링크를 통해 전송해야 할 패킷이 있는 사용자들 사이에서만 공유됩니다.

오늘날의 통신 네트워크에서는 패킷 스위칭과 회선 스위칭이 모두 사용되지만, 확실히 패킷 스위칭으로의 전환이 두드러지고 있습니다. 오늘날의 많은 회선 스위칭 전화 네트워크도 패킷 스위칭으로 천천히 전환되고 있습니다. 특히 전화 네트워크는 해외 전화의 고비용 부분에서 패킷 스위칭을 자주 사용합니다.

1.3.3 네트워크의 네트워크

앞서 엔드 시스템(PC, 스마트폰, 웹 서버, 메일 서버 등)이 액세스 ISP를 통해 인터넷에 연결된다는 것을 보았습니다. 액세스 ISP는 DSL, 케이블, FTTH, Wi-Fi, 셀룰러 등 다양한 액세스 기술을 사용해 유선 또는 무선 연결을 제공할 수 있습니다. 액세스 ISP는 반드시 통신사나 케이블 회사가 아니어도 됩니다. 예를 들어, 대학(학생, 직원, 교수에게 인터넷 액세스 제공)이나 회사(직원에게 액세스 제공)도 될 수 있습니다. 하지만 엔드 사용자와 콘텐츠 제공자를 액세스 ISP에 연결하는 것은 인터넷을 구성하는 수십억 개의 엔드 시스템을 연결하는 퍼즐의 작은 조각일 뿐입니다. 이 퍼즐을 완성하려면 액세스 ISP들 자체가 상호 연결되어야 합니다. 이를 위해 네트워크의 네트워크를 만드는 것이 필요하며, 이 개념을 이해하는 것이 인터넷을 이해하는 열쇠입니다.

수년에 걸쳐 인터넷을 형성하는 네트워크의 네트워크는 매우 복잡한 구조로 진화했습니다. 이 진화는 성능 고려사항보다는 경제적 요인과 국가 정책에 의해 주도되었습니다. 오늘날의 인터넷 네트워크 구조를 이해하기 위해, 점진적으로 네트워크 구조를 구축하며, 각 새로운 구조가 오늘날의 복잡한 인터넷에 더 가까운 근사치를 제공하도록 하겠습니다. 모든 엔드 시스템이 서로 패킷을 보낼 수 있도록 액세스 ISP를 상호 연결하는 것이 궁극적인 목표임을 기억하십시오. 한 가지 단순한 접근법은 각 액세스 ISP가 다른 모든 액세스 ISP와 직접 연결되는 것입니다. 하지만 이런 메쉬 설계는 액세스 ISP에게 너무 비용이 많이 들며, 전 세계 수십만 개의 다른 액세스 ISP와 각각 별도의 통신 링크를 가져야 하기 때문입니다.

첫 번째 네트워크 구조인 네트워크 구조 1은 모든 액세스 ISP를 단일 글로벌 트랜짓 ISP로 상호 연결합니다. (가상의) 글로벌 트랜짓 ISP는 전 세계를 아우르며, 수십만 개의 액세스 ISP 근처에 최소한 하나의 라우터를 가진 라우터와 통신 링크 네트워크입니다. 물론 글로벌 ISP가 이렇게 광범위한 네트워크를 구축하는 것은 매우 비용이 많이 듭니다. 수익을 내기 위해 글로벌 ISP는 각 액세스 ISP에게 연결 비용을 청구하며, 이 가격은 액세스 ISP가 글로벌 ISP와 교환하는 트래픽 양을 반영합니다(꼭 비례할 필요는 없음). 액세스 ISP는 글로벌 트랜짓 ISP에 비용을 지불하므로, 액세스 ISP는 고객이고 글로벌 트랜짓 ISP는 제공자라고 합니다.

어떤 회사가 수익성 있는 글로벌 트랜짓 ISP를 구축하고 운영한다면, 다른 회사들도 자체 글로벌 트랜짓 ISP를 구축하여 원래의 글로벌 트랜짓 ISP와 경쟁하는 것이 자연스럽습니다. 이는 네트워크 구조 2로 이어지며, 이 구조는 수십만 개의 액세스 ISP와 여러 글로벌 트랜짓 ISP로 구성됩니다. 액세스 ISP는 가격과 서비스에 따라 경쟁하는 글로벌 트랜짓 제공자를 선택할 수 있으므로 네트워크 구조 1보다 네트워크 구조 2를 선호합니다. 하지만 글로벌 트랜짓 ISP들 자체도 상호 연결되어야 합니다. 그렇지 않으면 한

글로벌 트랜짓 제공자에 연결된 액세스 ISP는 다른 글로벌 트랜짓 제공자에 연결된 액세스 ISP와 통신할 수 없습니다.

방금 설명한 네트워크 구조 2는 글로벌 트랜짓 제공자가 최상위 계층에, 액세스 ISP가 최하위 계층에 있는 2계층 계층 구조입니다. 이는 글로벌 트랜짓 ISP가 모든 액세스 ISP에 근접할 수 있을 뿐만 아니라 그렇게 하는 것이 경제적으로 바람직하다고 가정합니다. 실제로는 일부 ISP가 인상적인 글로벌 커버리지를 가지고 많은 액세스 ISP와 직접 연결하지만, 전 세계 모든 도시에 존재하는 ISP는 없습니다. 대신, 특정 지역에는 해당 지역의 액세스 ISP가 연결되는 지역 ISP가 있을 수 있습니다. 각 지역 ISP는 Tier-1 ISP에 연결됩니다. Tier-1 ISP는 (가상의) 글로벌 트랜짓 ISP와 유사하지만, 실제로 존재하는 Tier-1 ISP는 전 세계 모든 도시에 존재하지 않습니다. Level 3 Communications, AT&T, Sprint, NTT 등 약 12개의 Tier-1 ISP가 있습니다. 흥미롭게도 Tier-1 상태는 공식적으로 승인되지 않으며, 속담처럼 “그룹의 일원인지 물어야 한다면, 당신은 아마도 아닐 것”입니다.

네트워크의 네트워크로 돌아가서, 여러 경쟁 Tier-1 ISP뿐만 아니라 지역 내 여러 경쟁 지역 ISP도 있을 수 있습니다. 이러한 계층 구조에서 각 액세스 ISP는 연결된 지역 ISP에 비용을 지불하고, 각 지역 ISP는 연결된 Tier-1 ISP에 비용을 지불합니다. (액세스 ISP는 Tier-1 ISP에 직접 연결할 수도 있으며, 이 경우 Tier-1 ISP에 비용을 지불합니다.) 따라서 계층 구조의 각 수준에서 고객-제공자 관계가 존재합니다. Tier-1 ISP는 계층 구조의 최상위에 있으므로 누구에게도 비용을 지불하지 않습니다. 상황을 더 복잡하게 만드는 것은, 일부 지역에서는 더 큰 지역 ISP(전체 국가를 아우를 수 있음)가 존재하며, 해당 지역의 더 작은 지역 ISP가 이에 연결되고, 더 큰 지역 ISP는 Tier-1 ISP에 연결됩니다. 예를 들어, 중국에서는 각 도시에 액세스 ISP가 있고, 이는 주 ISP에 연결되며, 주 ISP는 국가 ISP에, 국가 ISP는 최종적으로 Tier-1 ISP에 연결됩니다 [Tian 2012]. 이 다중 계층 구조는 오늘날의 인터넷에 대한 대략적인 근사치일 뿐이며, 이를 네트워크 구조 3이라고 부릅니다.

오늘날의 인터넷에 더 가까운 네트워크를 구축하려면 계층적 네트워크 구조 3에 PoP(Points of Presence), 멀티-호밍, 피어링, **인터넷 교환 지점(IXP)**을 추가해야 합니다. PoP는 최하위(액세스 ISP) 수준을 제외한 모든 계층에 존재합니다. PoP는 제공자 네트워크 내에서 고객 ISP가 제공자 ISP에 연결할 수 있는 하나 이상의 라우터 그룹(동일 위치)입니다. 고객 네트워크가 제공자의 PoP에 연결하려면 제3자 통신 제공자로부터 고속 링크를 임대하여 고객의 라우터를 PoP의 라우터에 직접 연결할 수 있습니다. Tier-1 ISP를 제외한 모든 ISP는 멀티-호밍을 선택할 수 있습니다. 즉, 두 개 이상의 제공자 ISP에 연결할 수 있습니다. 예를 들어, 액세스 ISP는 두 개의 지역 ISP에 멀티-호밍하거나, 두 개의 지역 ISP와 Tier-1 ISP에 멀티-호밍할 수 있습니다. 마찬가지로 지역 ISP는 여러 Tier-1 ISP에 멀티-호밍할 수 있습니다. ISP가 멀티-호밍하면, 한 제공자가 실패하더라도 인터넷으로 패킷을 계속 보내고 받을 수 있습니다.

고객 ISP는 글로벌 인터넷 상호 연결성을 얻기 위해 제공자 ISP에 비용을 지불합니다. 고객 ISP가 제공자 ISP에 지불하는 금액은 제공자와 교환하는 트래픽 양을 반영합니다. 이러한 비용을 줄이기 위해, 계층 구조에서 동일한 수준의 근접한 두 ISP는 피어링을 할 수 있습니다. 즉, 두 ISP는 네트워크를 직접 연결하여 상위 중개자를 통하지 않고 모든 트래픽을 직접 연결을 통해 전달할 수 있습니다. 두 ISP가 피어링할 때는 일반적으로 **정산 무료(settlement-free)**이며, 어느 쪽도 상대방에게 비용을 지불하지 않습니다. 앞서 언급했듯이, Tier-1 ISP도 서로 정산 무료로 피어링합니다. 피어링과 고객-제공자 관계에 대한 읽기 쉬운 논의는 [Van der Berg 2008]에서 확인할 수 있습니다. 이와 비슷하게, 제3자 회사는 여러 ISP가 함께 피어링할 수 있는 **인터넷 교환 지점(IXP)**을 만들 수 있습니다. IXP는 일반적으로 자체 스위치를 가진 독립 건물에 위치합니다 [Ager 2012]. 현재

인터넷에는 600개 이상의 IXP가 있습니다 [PeeringDB 2020]. 액세스 ISP, 지역 ISP, Tier-1 ISP, PoP, 멀티-호밍, 피어링, IXP로 구성된 이 생태계를 네트워크 구조 4라고 부릅니다.

이제 오늘날의 인터넷을 설명하는 네트워크 구조 5에 도달합니다. 네트워크 구조 5는 그림 1.15에 나와 있으며, 네트워크 구조 4에 콘텐츠 제공자 네트워크를 추가하여 구축됩니다. 현재 구글은 콘텐츠 제공자 네트워크의 대표적인 예입니다. 현재 구글은 북아메리카, 유럽, 아시아, 남아메리카, 호주에 걸쳐 19개의 주요 데이터 센터를 보유하고 있으며, 각 데이터 센터에는 수만에서 수십만 대의 서버가 있습니다. 또한 구글은 수백 대의 서버를 가진 더 작은 데이터 센터를 보유하고, 이러한 소규모 데이터 센터는 종종 IXP 내에 위치합니다. 구글 데이터 센터는 전 세계를 아우르는 구글의 전용 TCP/IP 네트워크를 통해 상호 연결되지만, 공용 인터넷과는 분리되어 있습니다. 중요한 점은 구글 전용 네트워크는 구글 서버로/로부터의 트래픽만 전달한다는 것입니다. 그림 1.15에 나와 있듯이, 구글 전용 네트워크는 하위 계층 ISP와 직접 연결하거나 IXP에서 연결함으로써 인터넷의 상위 계층을 "우회"하려고 시도합니다 [Labovitz 2010]. 하지만 많은 액세스 ISP는 여전히 Tier-1 네트워크를 통과해야만 도달할 수 있으므로, 구글 네트워크는 Tier-1 ISP에도 연결되며, 이들과 교환하는 트래픽에 대해 비용을 지불합니다. 자체 네트워크를 구축함으로써 콘텐츠 제공자는 상위 계층 ISP에 지불하는 비용을 줄일 뿐만 아니라, 서비스가 최종 사용자에게 전달되는 방식을 더 많이 제어할 수 있습니다. 구글의 네트워크 인프라는 2.6절에서 더 자세히 설명됩니다.

PoP(Points of Presence): ISP 네트워크 내에서 고객 ISP가 연결할 수 있는 라우터 그룹. 동일 위치에 있으며, 고속 링크를 통해 연결을 제공한다.

멀티-호밍: ISP가 두 개 이상의 상위 ISP에 연결하여 안정성을 높이는 방식. 한 제공자 실패 시에도 인터넷 연결을 유지한다.

피어링: 동일 계층의 ISP가 직접 네트워크를 연결해 트래픽을 교환하는 것. 보통 정산 무료로 비용을 절감한다.

콘텐츠 제공자 네트워크: 구글과 같은 콘텐츠 제공자가 자체 네트워크를 구축해 트래픽을 관리. 하위 ISP나 IXP에 직접 연결해 비용과 지연을 줄인다.

PoP(Points of Presence): ISP 네트워크에서 고객 ISP가 연결하는 라우터 모음. 같은 장소에 있으며, 빠른 링크로 데이터 전송을 지원한다.

멀티-호밍: ISP가 여러 상위 ISP에 연결해 안정성을 높이는 방법. 한 제공자가 끊겨도 데이터 흐름이 유지된다.

피어링: 같은 수준의 ISP가 서로 네트워크를 직접 연결해 데이터 전송을 나누는 것. 보통 비용 없이 이루어진다.

콘텐츠 제공자 네트워크: 구글 같은 회사가 자체 네트워크를 만들어 데이터 흐름을 관리. 하위 ISP나 교환 지점에 직접 연결해 비용과 지연을 줄인다.

요약

오늘날의 인터넷—네트워크의 네트워크—is 약 12개의 Tier-1 ISP와 수십만 개의 하위 계층 ISP로 구성된 복잡한 구조입니다. ISP는 커버리지 면에서 다양하며, 일부는 여러 대륙과 대양을 아우르고, 다른 일부는 좁은 지역에 국한됩니다. 하위 계층 ISP는 상위 계층 ISP에

연결되고, 상위 계층 ISP는 서로 상호 연결됩니다. 사용자와 콘텐츠 제공자는 하위 계층 ISP의 고객이며, 하위 계층 ISP는 상위 계층 ISP의 고객입니다. 최근 몇 년 동안 주요 콘텐츠 제공자는 자체 네트워크를 구축하여 가능한 곳에서는 하위 계층 ISP에 직접 연결하고 있습니다.

1.4 패킷 스위칭 네트워크에서의 지연, 손실, 처리량

1.1절에서 우리는 인터넷을 엔드 시스템에서 실행되는 분산 애플리케이션에 서비스를 제공하는 인프라로 볼 수 있다고 언급했습니다. 이상적으로 우리는 인터넷 서비스가 두 엔드 시스템 간에 원하는 만큼의 데이터를 즉시, 데이터 손실 없이 이동할 수 있기를 바랍니다. 그러나 이는 현실적으로 달성 불가능한 높은 목표입니다. 대신, 컴퓨터 네트워크는 엔드 시스템 간의 처리량(초당 전송할 수 있는 데이터의 양)을 제약하고, 엔드 시스템 간에 지연을 발생시키며, 심지어 패킷 손실을 초래할 수 있습니다. 한편으로는 물리적 현실의 법칙이 지연과 손실을 발생시키고 처리량을 제약하는 것이 유감스럽습니다. 다른 한편으로는, 컴퓨터 네트워크가 이러한 문제를 가지기 때문에 이를 해결하는 방법과 관련된 흥미로운 문제들이 많아 컴퓨터 네트워킹 강의를 채우고 수천 개의 박사 논문을 동기부여하기에 충분합니다! 이 섹션에서는 컴퓨터 네트워크에서의 지연, 손실, 처리량을 조사하고 정량화하기 시작할 것입니다.

1.4.1 패킷 스위칭 네트워크에서의 지연 개요

패킷은 호스트(소스)에서 시작되어 일련의 라우터를 거쳐 다른 호스트(목적지)에서 여정을 마무리합니다. 패킷이 이 경로를 따라 한 노드(호스트 또는 라우터)에서 다음 노드(호스트 또는 라우터)로 이동할 때, 경로상의 각 노드에서 여러 유형의 지연을 겪습니다. 가장 중요한 지연은 노드 처리 지연, 큐잉 지연, 전송 지연, 전파 지연이며, 이 지연들이 누적되어 총 노드 지연을 구성합니다. 검색, 웹 브라우징, 이메일, 지도, 인스턴트 메시징, VoIP와 같은 많은 인터넷 애플리케이션의 성능은 네트워크 지연에 크게 영향을 받습니다. 패킷 스위칭과 컴퓨터 네트워크를 깊이 이해하려면 이러한 지연의 본질과 중요성을 이해해야 합니다.

지연의 유형

그림 1.16의 맥락에서 이러한 지연을 살펴보겠습니다. 소스와 목적지 간의 엔드-투-엔드 경로의 일부로, 패킷은 상위 노드에서 라우터 A를 거쳐 라우터 B로 전송됩니다. 우리의 목표는 라우터 A에서의 노드 지연을 특성화하는 것입니다. 라우터 A는 라우터 B로 이어지는 출력 링크를 가지고 있습니다. 이 링크 앞에는 큐(버퍼라고도 함)가 있습니다. 패킷이 상위 노드에서 라우터 A에 도착하면, 라우터 A는 패킷의 헤더를 검사하여 패킷에 적합한 출력 링크를 결정하고 패킷을 이 링크로 보냅니다. 이 예에서 패킷의 출력 링크는 라우터 B로 이어지는 링크입니다. 패킷은 링크에서 현재 다른 패킷이 전송되고 있지 않고, 큐에 앞서 대기 중인 다른 패킷이 없는 경우에만 해당 링크를 통해 전송될 수 있습니다. 링크가 현재 사용 중이거나 큐에 이미 다른 패킷이 대기 중이라면, 새로 도착한 패킷은 큐에 추가됩니다.

처리 지연

패킷의 헤더를 검사하고 패킷을 어디로 보낼지 결정하는 데 필요한 시간은 처리 지연의 일부입니다. 처리 지연에는 패킷의 비트가 상위 노드에서 라우터 A로 전송되는 동안 발생하는 비트 단위 오류를 확인하는 데 필요한 시간과 같은 다른 요소도 포함될 수 있습니다. 고속 라우터에서의 처리 지연은 일반적으로 마이크로초 이하 수준입니다. 이 노드 처리가 완료된

후, 라우터는 패킷을 라우터 B로 이어지는 링크 앞의 큐로 보냅니다. (4장에서 라우터의 작동 방식에 대한 세부 사항을 살펴볼 것입니다.)

큐잉 지연

큐에서 패킷은 링크로 전송되기를 기다리며 큐잉 지연을 겪습니다. 특정 패킷의 큐잉 지연의 길이는 큐에 이미 대기 중이며 링크로 전송을 기다리는 이전에 도착한 패킷의 수에 따라 달라집니다. 큐가 비어 있고 현재 전송 중인 다른 패킷이 없다면, 우리 패킷의 큐잉 지연은 0이 됩니다. 반면, 트래픽이 많아 많은 패킷이 전송을 기다리고 있다면, 큐잉 지연은 길어질 것입니다. 곧 보겠지만, 도착 패킷이 큐에서 마주칠 것으로 예상되는 패킷 수는 큐에 도착하는 트래픽의 강도와 특성에 따라 달라집니다. 실제로 큐잉 지연은 마이크로초에서 밀리초 단위일 수 있습니다.

전송 지연

패킷이 선입선출(FIFO) 방식으로 전송된다고 가정하면, 이는 패킷 스위칭 네트워크에서 일반적입니다. 이 경우 우리 패킷은 이전에 도착한 모든 패킷이 전송된 후에야 전송될 수 있습니다. 패킷의 길이를 L 비트, 라우터 A에서 라우터 B로의 링크 전송 속도를 R 비트/초라고 하겠습니다. 예를 들어, 10Mbps 이더넷 링크의 경우 속도는 $R = 10\text{Mbps}$ 이고, 100Mbps 이더넷 링크의 경우 속도는 $R = 100\text{Mbps}$ 입니다. 전송 지연은 L/R 입니다. 이는 패킷의 모든 비트를 링크로 내보내는(즉, 전송하는) 데 필요한 시간입니다. 실제로 전송 지연은 마이크로초에서 밀리초 단위일 수 있습니다.

전파 지연

비트가 링크로 내보내지면, 라우터 B로 전파되어야 합니다. 링크의 시작에서 라우터 B까지 전파되는 데 필요한 시간을 전파 지연이라고 합니다. 비트는 링크의 전파 속도로 전파됩니다. 전파 속도는 링크의 물리적 매체(예: 광섬유, 꼬인 쌍 구리선 등)에 따라 달라지며, 다음과 같은 범위에 있습니다:

$2 * 10^8 \text{ m/s}$ 에서 $3 * 10^8 \text{ m/s}$

이는 빛의 속도와 같거나 약간 느립니다. 전파 지연은 두 라우터 간의 거리를 전파 속도로 나눈 값입니다. 즉, 전파 지연은 d/s 이며, 여기서 d 는 라우터 A와 라우터 B 간의 거리이고, s 는 링크의 전파 속도입니다. 패킷의 마지막 비트가 노드 B로 전파되면, 해당 비트와 패킷의 모든 이전 비트가 라우터 B에 저장됩니다. 이후 전체 프로세스는 라우터 B가 이제 포워딩을 수행하면서 계속됩니다. 광역 네트워크에서는 전파 지연이 밀리초 단위일 수 있습니다.

전송 지연과 전파 지연 비교

컴퓨터 네트워킹을 처음 접하는 사람들은 종종 전송 지연과 전파 지연의 차이를 이해하는 데 어려움을 겪습니다. 이 차이는 미묘하지만 중요합니다. 전송 지연은 라우터가 패킷을 내보내는 데 필요한 시간으로, 패킷의 길이와 링크의 전송 속도에 따라 달라지며, 두 라우터 간의 거리와는 무관합니다. 반면, 전파 지연은 비트가 한 라우터에서 다음 라우터로 전파되는 데 걸리는 시간으로, 두 라우터 간의 거리에 따라 달라지며, 패킷의 길이나 링크의 전송 속도와는 무관합니다.

비유를 통해 전송 지연과 전파 지연의 개념을 명확히 해보겠습니다. 그림 1.17과 같이, 100km마다 톨게이트가 있는 고속도로를 생각해 봅시다. 톨게이트 사이의 고속도로 구간을 링크로, 톨게이트를 라우터로 간주할 수 있습니다. 자동차가 고속도로에서 100km/h의

속도로 이동(즉, 전파)한다고 가정합니다(즉, 자동차가 톨게이트를 떠나면 즉시 100km/h로 가속하여 톨게이트 사이에서 그 속도를 유지합니다). 또한, 10대의 자동차가 캐러밴으로 함께 고정된 순서로 이동한다고 가정합니다. 각 자동차를 비트로, 캐러밴을 패킷으로 생각할 수 있습니다. 각 톨게이트는 자동차를 12초당 한 대씩 처리(즉, 전송)한다고 가정하며, 늦은 밤이라 캐러밴의 자동차들이 고속도로에 있는 유일한 자동차라고 가정합니다. 마지막으로, 캐러밴의 첫 번째 자동차가 톨게이트에 도착하면, 나머지 9대의 자동차가 도착하여 뒤에 줄을 설 때까지 입구에서 대기한다고 가정합니다. (따라서 캐러밴 전체가 톨게이트에 저장된 후에야 포워딩이 시작됩니다.)

톨게이트가 캐러밴 전체를 고속도로로 내보내는 데 필요한 시간은 $(10 \text{ 대}) / (5 \text{ 대/분}) = 2$ 분입니다. 이 시간은 라우터의 전송 지연에 해당합니다. 자동차가 한 톨게이트의 출구에서 다음 톨게이트까지 이동하는 데 필요한 시간은 $100 \text{ km} / (100 \text{ km/h}) = 1$ 시간입니다. 이 시간은 전파 지연에 해당합니다. 따라서 캐러밴이 한 톨게이트 앞에 저장된 시점부터 다음 톨게이트 앞에 저장되는 시점까지의 시간은 전송 지연과 전파 지연의 합으로, 이 예에서는 62분입니다.

이 비유를 조금 더 살펴보겠습니다. 톨게이트의 캐러밴 처리 시간이 톨게이트 간 이동 시간보다 길다면 어떻게 될까요? 예를 들어, 자동차가 1,000km/h로 이동하고 톨게이트가 1분당 한 대의 자동차를 처리한다고 가정합시다. 그러면 톨게이트 간 이동 지연은 6분이고, 캐러밴을 처리하는 데 10분이 걸립니다. 이 경우, 캐러밴의 첫 몇 대의 자동차는 캐러밴의 마지막 자동차가 첫 번째 톨게이트를 떠나기 전에 두 번째 톨게이트에 도착할 것입니다. 이 상황은 패킷 스위칭 네트워크에서도 발생합니다. **패킷의 첫 번째 비트는 패킷의 나머지 비트가 이전 라우터에서 아직 전송 대기 중일 때 다음 라우터에 도착할 수 있습니다.**

그림이 천 마디 말보다 낫다면, 애니메이션은 백만 마디 말보다 나을 것입니다. 이 교재의 웹사이트는 전송 지연과 전파 지연을 훌륭하게 보여주고 비교하는 인터랙티브 애니메이션을 제공합니다. 독자는 이 애니메이션을 방문해 보는 것을 강력히 권장합니다. [Smith 2009]는 전파, 큐잉, 전송 지연에 대한 매우 읽기 쉬운 논의를 제공합니다.

노드 지연의 총합은 다음과 같이 표현됩니다:

$$d_{\text{nodal}} = d_{\text{proc}} + d_{\text{queue}} + d_{\text{trans}} + d_{\text{prop}}$$

(설명: 여기서 d_{nodal} 은 초 단위의 총 노드 지연, d_{proc} 은 처리 지연, d_{queue} 는 큐잉 지연, d_{trans} 는 전송 지연, d_{prop} 은 전파 지연입니다.)

이 지연 구성 요소들의 기여도는 상황에 따라 크게 달라질 수 있습니다. 예를 들어, d_{prop} 은 같은 대학 캠퍼스 내의 두 라우터를 연결하는 링크에서는 무시할 만할 수 있습니다(예: 몇 마이크로초). 하지만 정지계도 위성 링크로 연결된 두 라우터에서는 수백 밀리초에 달하며, d_{nodal} 에서 지배적인 요소가 될 수 있습니다. 마찬가지로, d_{trans} 는 10Mbps 이상의 전송 속도(예: LAN)에서는 무시할 만하지만, 저속 다이얼업 모뎀 링크를 통해 큰 인터넷 패킷을 보낼 때는 수백 밀리초에 달할 수 있습니다. 처리 지연 d_{proc} 은 종종 무시할 만하지만, 라우터의 최대 처리량(라우터가 패킷을 포워딩할 수 있는 최대 속도)에 강하게 영향을 미칩니다.

1.4.2 큐잉 지연과 패킷 손실

노드 지연의 가장 복잡하고 흥미로운 구성 요소는 큐잉 지연 d_{queue} 입니다. 사실, 큐잉 지연은 컴퓨터 네트워킹에서 매우 중요하고 흥미로워서 이에 관한 수천 편의 논문과 수많은

책이 작성되었습니다 [Bertsekas 1991; Kleinrock 1975, Kleinrock 1976]. 여기서는 큐잉 지연에 대한 고수준의 직관적인 논의를 제공하며, 더 궁금한 독자는 관련 책을 살펴보거나 (심지어 박사 논문을 작성할 수도 있습니다!). 다른 세 가지 지연(d_{proc} , d_{trans} , d_{prop})과 달리, 큐잉 지연은 패킷마다 달라질 수 있습니다. 예를 들어, 10개의 패킷이 빈 큐에 동시에 도착하면, 첫 번째로 전송되는 패킷은 큐잉 지연을 전혀 겪지 않지만, 마지막으로 전송되는 패킷은 상대적으로 큰 큐잉 지연을 겪습니다(다른 9개의 패킷이 전송될 때까지 기다리므로). 따라서 큐잉 지연을 특성화할 때는 평균 큐잉 지연, 큐잉 지연의 분산, 큐잉 지연이 특정 값을 초과할 확률과 같은 통계적 척도를 일반적으로 사용합니다.

큐잉 지연은 언제 크고, 언제 무시할 만한가요? 이 질문에 대한 답은 큐에 도착하는 트래픽의 속도, 링크의 전송 속도, 그리고 도착 트래픽의 특성(즉, 트래픽이 주기적으로 도착하는지, 버스트 형태로 도착하는지)에 따라 달라집니다. 이에 대한 통찰을 얻기 위해, 큐에 패킷이 도착하는 평균 속도를 a (단위: 패킷/초)라고 합시다. R 은 전송 속도, 즉 비트가 큐에서 내보내지는 속도(단위: 비트/초)입니다. 간단히 가정하기 위해 모든 패킷이 L 비트로 구성되어 있다고 합시다. 그러면 큐에 도착하는 비트의 평균 속도는 La 비트/초입니다. 또한, 큐가 매우 커서 사실상 무한한 비트를 저장할 수 있다고 가정합니다. 트래픽 강도라고 불리는 La/R 은 큐잉 지연의 정도를 추정하는 데 중요한 역할을 합니다. 만약 $La/R > 1$ 이라면, 큐에 도착하는 비트의 평균 속도가 큐에서 비트를 전송할 수 있는 속도를 초과합니다. 이 불행한 상황에서는 큐가 무한히 증가하는 경향이 있으며, 큐잉 지연은 무한대에 가까워집니다! 따라서 트래픽 엔지니어링의 황금을 중 하나는 다음과 같습니다: 트래픽 강도가 1을 초과하지 않도록 시스템을 설계하라.

이제 $La/R \leq 1$ 인 경우를 살펴보겠습니다. 여기서 도착 트래픽의 특성이 큐잉 지연에 영향을 미칩니다. 예를 들어, 패킷이 주기적으로 도착한다면(즉, L/R 초마다 한 패킷이 도착), 모든 패킷은 빈 큐에 도착하므로 큐잉 지연이 없습니다. 반면, 패킷이 버스트 형태로 주기적으로 도착하면 평균 큐잉 지연이 상당할 수 있습니다. 예를 들어, $(L/R)N$ 초