CHAPTER 1

TECHNOLOGIES SUPPORTING MOBILE DATA

1.1 INTRODUCTION

The popularity of mobile devices is growing exponentially, and the number of such devices deployed worldwide is rising rapidly. These devices come in many forms such as Smartphones, Personal Digital Assistants, and tablets. Another important category of mobile devices are laptops where the use of network interfaces that support mobile wireless data, either via cellular data networks or through Wi-Fi, is commonplace.

The increase in number of mobile devices has led to the development of many new applications targeted at both businesses and consumers. The count of such applications runs into several hundreds of thousands for any of the popular mobile device platforms. Some of these applications require very little network data exchange but some, for example, applications which allow a user to watch streaming video on the mobile device can use up a tremendous amount of data in a very short amount of time.

As a result, the amount of data that is sent over the mobile cellular networks keeps on increasing steadily. The amount of data growth in mobile networks is tracked by various organizations. Studies conducted by several companies [1,2] indicate that mobile data growth has approximately been tripling every year since 2009. A significant growth in the data came from the

Techniques for Surviving the Mobile Data Explosion, First Edition.
Dinesh Chandra Verma and Paridhi Verma.

transmission of video on the network, which accounts for more than half of the total mobile data. Furthermore, predictions made by both these studies, as well as other studies, show no signs of slowing down of this trend of mobile data growth.

The bandwidth demands from all of the mobile applications that exist currently and will be developed in the future will likely exceed the network capacity that can be offered by the currently deployed wireless cellular networks. This mismatch in capacity can be handled in a variety of ways, and the different techniques that can be used to solve the capacity mismatch are the subject of this book.

In an ideal world, one would simply upgrade the network infrastructure so that all of the challenges associated with limited bandwidth disappeared. In real life, this simple solution comes with an enormous price tag. There are several constituencies with some vested interest in mobile data communications, for example, mobile device users, mobile network operators, mobile application developers, enterprises using mobile computing applications. Each of these constituencies would like the bulk of cost required for bandwidth upgrade to be borne by some other constituency. The actions to address the capacity mismatch that are within the control of each constituency are also different. An introduction to the various constituencies in the mobile ecosystem and the implications of mobile data growth on each of the constituencies is provided in Chapter 2 of this book.

Approaches that can be used to address mobile data growth have technical complexities in addition to the ecosystem complexities. Mobile data communications are at the intersection of three different, though related, technical fields, namely (i) mobile applications, (ii) Internet, and (iii) cellular networks. Approaches to address mobile data growth need to span three technical areas, which is more complex than any approach that could be addressed within one single technical area.

The relationship and interaction between these three areas can be understood by a reference to Figure 1.1. The figure shows a high-level layout

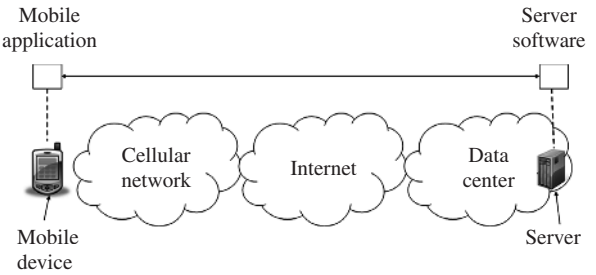

FIGURE 1.1 Mobile applications data communication infrastructure.

of the communication infrastructure needed for a mobile application that is exchanging data with another computer on the Internet.

Mobile applications are software components that run on mobile devices such as a Smartphone or tablet computer. They exchange data with application components running on servers, shown as server software in Figure 1.1. In order to communicate with each other, the mobile application and server software use a set of conventions called a communication protocol. The Hypertext Transport Protocol (HTTP) is an example of a common protocol used in this manner. The information exchanged by this protocol traverses a number of networks between the mobile device and the server. These networks include a cellular network that connects the mobile device wirelessly to the Internet, and the Internet that provides a means to connect the cellular network to the data center where the server may be physically located.

As the figure demonstrates, mobile applications run using communication protocols overlaid on top of Internet protocols (IPs) that are overlaid partially on top of cellular networks. Each of these three technical fields has its own sophisticated set of well-developed technologies and best practices. Two of these technical fields, namely cellular networks and Internet are specific cases of computer communication networks and have some terminology and design principles in common. However, due to historical reasons, the Internet and the cellular networks have each evolved terminology and mechanisms that are very different. Managing the growth of data due to mobile applications requires an approach that can span across each of these three technical fields and take into account the idiosyncrasies and characteristics of each of these fields.

In this introductory chapter, we provide a high level overview of these technology areas, beginning with an overview of data networks, followed by a discussion about the Internet (an instance of a data network) and the cellular networks (another instance of a data network), and the mobile applications protocols.

1.2 COMPUTER COMMUNICATION NETWORKS

The basic function of a computer communication network is to enable two or more computers to exchange data with each other. To enable this exchange, the computers need to agree to a set of conventions that they can all understand and agree upon. Such a set of conventions is called a communication protocol. Computer communication networks consist of a set of communication protocols that build upon each other in a layered manner.

In order for two computers that may be in two different continents to communicate, several layers of communication protocols are needed. Let us consider a

simple interaction, one that happens when a user types in the address of a website like http://www.chandabooks.com in a web browser on his Smartphone, and another computer sends back some data which is displayed to the user. In order to facilitate this exchange, a protocol called HTTP is used between the Smartphone and the IEEE web server. The protocol defines the formats in which the requests from the Smartphone to the web server ought to be sent, and how the web server will respond back to it. This protocol is built using the assumption that all requests and responses are received by the other party reliably.

On any real network, requests and responses may be lost in transit. In order to not worry about such losses, HTTP protocol is usually layered on top of another protocol called Transmission Control Protocol (TCP). This protocol has conventions that let the other computer know whether data has been received reliably, how to detect if some data was lost and how to retransmit it, and how the receiving computer can put data back in order the sender sent it even if some data was received out of order. Note that HTTP does not have to be tied with TCP. It could run equally well on top of any other protocol that provided reliable in-order communication. Transmission Control Protocol in turn would be layered on another protocol, for example, the IP. The IP itself in turn would be layered on another set of protocols.

Most data networks are designed with several layers on top of each other, each layer consisting of one or more protocols. Classical data networking texts usually begin with an overview of a canonical protocol stack consisting of seven layers. However, the seven layers are rarely implemented in real life, so the canonical protocol stack is only of academic interest. However, terminology from the canonical protocol stack may frequently be encountered in technical networking literature. We will point out that terminology when appropriate in the rest of this chapter. The only point from the seven-layer canonical model that we will mention is that the model starts its numbering from the bottom, so that in the example of HTTP and TCP that we gave, the HTTP would correspond to a layer that is numerically higher than the layer of TCP.

It would be easier to understand modern practical computer communication protocols in terms of four layers that are shown in Figure 1.2. The bottommost layer, marked as LINK/MAC, refers to a protocol that will allow two computers to communicate with each other as long as they are physically connected to each other, for example, if there is a wire connecting them or they are within the range of wireless communication with each other. This protocol would typically correspond to layers one and two of the canonical seven-layer model. The second layer, NETWORK, refers to a protocol that will allow computers that are connected indirectly via a set of one or more

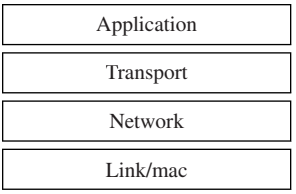

FIGURE 1.2 Data networking layers.

networks sharing a LINK/MAC layer with each other. The third layer, TRANSPORT, refers to a protocol that runs end-to-end between the two computers that are communicating across the network and allows information delivery while coping with issues such as reliability of transmission, ensuring information is delivered in sequence, and that different computers needing to communicate do not overwhelm resources in the network. The fourth layer, APPLICATION, covers any other end-to-end protocol that is required to enable communication atop transport.

At the network layer, protocols tend to fall into two major categories, circuit-switched and packet-switched. Circuit-switched protocols owe their heritage to the telephone networks, when in the old days human operators would plug-in connectors at different telephone exchanges to establish a dedicated line between two people making a phone call. In data networks using circuit-switching, a similar mechanism reserves resources for each pair of communicating computers at any intermediaries to create a dedicated communication channel. This method ensures a good quality of service but is relatively inefficient in resource usage. In packet switching, information is carried into discrete information fragments called packets, each packet carrying a header which allows intermediary nodes to send packets to the right receiver. Due to its efficiencies, most data networks today use a packet-switching paradigm even though it may be subject to fluctuations in service quality when too many packets bunch up at a single location in the network. Some network protocols use the concept of virtual circuits where a logical end-to-end circuit is created on top of a packet-switching network, which provides a tradeoff between the characteristics of both paradigms.

In order for two computers to communicate across a data network, the four layers shown in Figure 1.2 would need to be supported in a manner as shown in Figure 1.3. Assume that computer A wants to exchange data with computer B through intervening computers C and D. The information exchange requires that A and B use the same application and transport protocols. Furthermore, all the four computers need to support the same network protocol. The network protocol allows the computers A and D to exchange data with each other even though the intervening computers may be connected using a different set of LINK/MAC technologies. In the example shown in Figure 1.3,

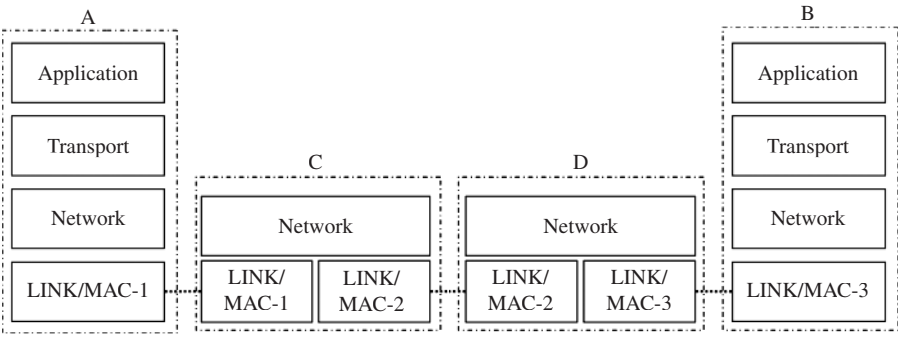

FIGURE 1.3 Data networking example.

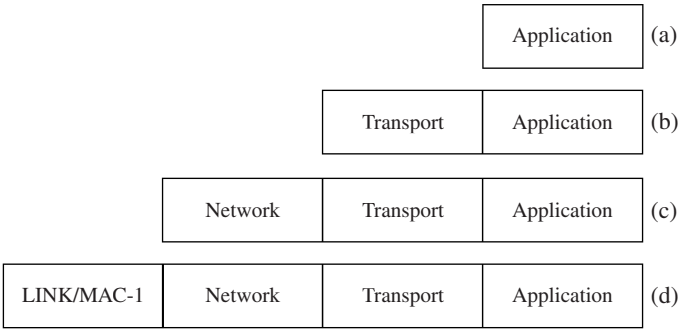

FIGURE 1.4 Data network packet structure.

each pair of computers A–C, C–D, and D–B are connected with a different type of LINK/MAC layer protocol.

When the computer A wants to send some application level data to computer B, a piece of software responsible for processing the application protocol on computer A will create that application level data. That data is marked as (a) in Figure 1.4. This data will then be passed to a piece of software on computer A responsible for transport layer processing. This software would attach some more information needed for transport protocol functions so that the data to be sent from A to B looks like the structure marked as (b) in Figure 1.4. The information needed for the transport layer is usually added at the front of the application level data and is called the transport header. The application level data is referred to as the payload. The transport header and the transport payload comprise the transport data. The transport data is then passed to a software responsible for network layer processing which attaches a network header making the data look like that of Figure 1.4(c). In this case, the transport data becomes the payload contained in the network data, which consists of a network header and the network payload (transport data).

Finally, the LINK/MAC layer-processing software will attach another header making the data look like that of Figure 1.4(d). Some portions of the preceding description may be done in hardware instead of software, but the end result is the same, producing the data format in Figure 1.4(d). This data format is what we will see if one were to take a snapshot of the data flowing from computer A to C.

Let us now examine what will happen to this packet when it reaches computer C. The LINK/MAC layer-processing software (or hardware) will strip off the LINK/MAC header recreating the data as it was in Figure 1.4(c). It will then be processed by the software responsible for the network layer at computer C, which may modify the network header. The structure would then be like that of Figure 1.4(c), but the contents of the network header could be modified and become different. When computer C sends this modified content to computer D, a new LINK/MAC header, this time corresponding to the LINK/MAC-2 protocol is attached to the data. The same sequence of steps is repeated at computer D, resulting in potential modification of network header, and the LINK/MAC header changes between every pair of computers.

The transport data is unchanged as it travels from computer A to B and processed by the transport software at computer B, which will extract the transport payload, that is, the original application data and deliver it to the software responsible for application-level processing at computer B.

The layering structure allows network communication at each layer to remain independent of the protocols used at layers above or below them. This has led to several creative combinations of network protocols to address different communication challenges that can arise in practical networks. Network protocols can be layered in many different ways, for example, an IP packet may be tunneled inside another IP packet to carry it across a network, or an IP packet may be layered on top of HTTP to cross firewall boundaries. The myriad of possible combinations of protocols has given rise to a very large and diverse set of protocol stacks.

1.3 IP NETWORKS

Although there have been many types of computer communications-networking technologies, the dominant technology in use today uses a protocol stack built upon IP. The most dominant version of this protocol that is almost universally deployed is IP Version 4, abbreviated usually as IPV4. Although there is a newer version IP Version 6 (IPV6) available in the market-place, it is yet to see significant adoption. The IPV4 protocol is the technical foundation on which the Internet and the World Wide Web (WWW) have been built. Mobile data applications are also usually based on IPV4

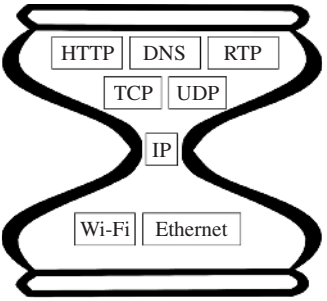

FIGURE 1.5 IP hour-glass structure.

technologies. In this book, we will simply use IP to refer to IPV4 protocol, calling out IPV6 explicitly when needed.

IP is a packet-switching network layer protocol which was developed to allow any two computers using the protocol to talk to each other, even if they happen to belong to different networks, that is, managed and administered by different organizations. Its primary purpose is to allow two networks, which may have many internal differences, to be able to exchange data effectively. In terms of the layering structure discussed in the previous section, IP would be a network layer protocol. However, as explained previously, it has been used in various other layers depending on the idiosyncrasies of specific environments.

Applications running on the IP protocol have a distribution of structure which is sometimes defined as the hourglass system, that is, if you draw a diagram of all the different types of protocols that are operational on the Internet, you will get an hourglass shape as shown in Figure 1.5. The IP protocol is at the narrow part of the hourglass. On top of it, there are a few commonly used protocols like TCP and Universal Datagram Protocol (UDP). More protocols are implemented on top of one (or both) of these protocols including the ubiquitous HTTP that defines the WWW implemented over TCP, the Domain Name Service (DNS) that is implemented on both TCP and UDP or Real-time Transfer Protocol (RTP) which is usually implemented over UDP. Above these protocols, more protocols are defined, for example, the Simple Object Access Protocol (SOAP) which defines web services is usually built atop HTTP. Application developers can create their own private protocols on top of any of the available protocols they find suitable for their purpose.

Below the IP layer are another suite of protocols which provide a way to interconnect a small set of computers. A common example is the Wi-Fi protocol, or more technically the IEEE 802.11 series of protocols. Wi-Fi is ubiquitous in homes, hotels, and airports. The Wi-Fi protocol allows a set of machines on a wireless network connected to a common access point to

communicate with each other. Ethernet is another common protocol that can connect a set of machines that are connected to an Ethernet switch. Consider the situation where a Wi-Fi access point is connected to an Ethernet switch and thus acts as a bridge between a Wi-Fi network and an Ethernet network. When a computer connected to the Wi-Fi network needs to communicate with a computer that is on the Ethernet network, they both need to use a common protocol that can help them talk to each other. The IP protocol provides this function.

The bottom half of the hourglass shape indicates that a wide variety of communication protocols can be developed within smaller networks that allow a group of computers to talk to each other. The cellular communication protocols provide a similar set of protocols underneath the hourglass of IP.

The Internet consists of all the computers and other devices supporting the IP protocol that are able to talk to each other. It is a collection of several IP-based networks, operated by many different organizations, which are connected to each other at a number of exchange locations or peering points.

In an IP network, each machine has a unique address, a 32-bit identifier in IPV4, which is used to address packets to that particular machine. The different nodes in the network talk to each other to find out the right way to route packets to the correct machine. This information exchange for routing packets happens at a slower rate than packets being generated, and one of the implicit assumptions required for the correct operation of IP networks is that a machine with an IP address does not move around rapidly within the network.

In addition to the addresses, some of the machines in the IP network also have a domain name. The domain name is a hierarchical name made up with human readable characters such as http://www.chandabooks.com. If the domain name of a machine is known, then any computer can determine its IP address using a distributed system called the domain name system. The domain name provides an easier way to identify a server which needs to be accessed by many applications.

Most of the communication in an IP network falls within the paradigm of client server computing. In this paradigm, a server is a machine whose domain name or IP address is well publicized. A client is a machine which initiates communication to the server, usually by looking up the domain name of the server to get its IP address, and then sending the first packet to the server using its IP address. Once the server receives the packet, it responds back to the client and they communicate according to the protocol stack they both have in common. In the specific context of mobile data communication, normally the mobile device is the client, and the other computer it gets data from, for example a website, is the server.

Having an indirection in the domain name of the server and the IP address allows some significant flexibility in communication as well. One can develop

schemes that map the same domain name to different IP addresses, for example, to a different address in each of the continents to have clients communicate to servers who are located in the same geographic neighborhood. This flexibility can be very useful to deal with the overload conditions arising due to the growth in data traffic, as explained in subsequent chapters.

One of the assumptions underlying the design of IP networks is that the location of any machine with an address within the network remains static. The manner in which packets are forwarded towards their destination reflects this design choice. There are extensions of IP communication which allow for some mobility of the machines, but despite their availability, IP remains primarily a protocol for networks where machines do not move around.

1.4 CELLULAR DATA NETWORKS

Cellular networks are also a type of computer communication network, but they evolved with the primary goal of supporting mobile users with wireless connections, very unlike IP networks which were focused on fixed computers. The cellular networks draw their name from the fact that they divide all of the area being serviced into disjoint regions called cells. Each cell is assigned a set of carrier frequencies, a portion of the electromagnetic spectrum that can be used by mobile devices in that cell to communicate with equipment in a tower servicing that cell. The frequencies can be reused in cells that are not adjacent to each other. Thus, a cellular structure allows coverage of a broad area using only a limited set of frequencies.

The tower servicing the cell can either be located in the center of the cell or on the corners of the cells. When the tower is in the center, it will use an antenna that transmits on all directions of it using the allocated frequencies. When the tower is at the edge, it will transmit on one of the allocated frequencies using a directional antenna, that is, an antenna that will transmit signals in a limited direction towards the interior of the cell. In this case, a single cell may be serviced by more than one tower.

Since users can move over from one cell to another, a system for managing hand-offs of users between cells is required in cellular networks. The specific details of how the hand-off happens depend on the communication technology which is used between a user's mobile device and the tower.

There are many protocols used within cellular networks, and some of the more common ones include General Packet Radio Service (GPRS), Universal Mobile Telecommunications System (UMTS), CDMA2000, IEEE 802.16 (Wi-Max), Long-Term Evolution (LTE), and LTE-Advanced. Each of these protocols requires a book to describe them properly. In order to not get bogged down in the specifics of these protocols, each of whom are very

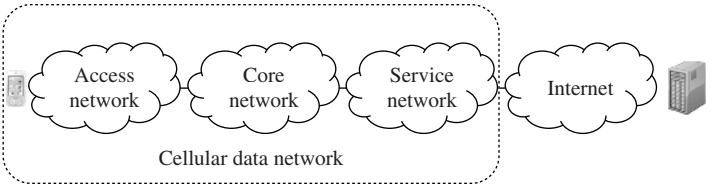

FIGURE 1.6 Components of a cellular data network.

complex, we will use a simplified description which will apply to all of these protocols in a generic manner.

The basic architecture of all types of cellular networks and the way in which they connect to the Internet can be viewed as consisting of three distinct but separate networks, an access network, a core network, and a service network. The service network will typically connect to the Internet through an IP router. Each of these components is shown in Figure 1.6.

The access network provides connectivity to the mobile devices and comprises of the equipment at the cell towers as well as other equipment required in the network for other functions, for example, controlling the operations of the radio network. Physically, the access network would consist of two distinct parts, one is the over-the-air segment which connects mobile devices to equipment at the cell tower and the other is the network which connects the cell-tower equipment to other equipment and eventually to the core network. The latter is usually referred to as the backhaul network. The backhaul network in some countries tends to be optical fiber in nature but in other countries consists of microwave.

The core network consists of devices that are responsible for authentication, access control, security, and mobility functions. As we have mentioned previously, IP networks are designed primarily to work with static machines. Within the core network, functionality is implemented so that this mobility becomes hidden from IP-based portion of the network. The exact mechanism to manage the mobility depends on the specific protocol running within the network, for example, CDMA2000 uses mobile-IP protocols to handle mobility while UMTS would tunnel packets from the mobile device to a few fixed locations in the core network called the Gateway GPRS Support Node (GGSN), in effect making the entire UMTS protocol act like a LINK/MAC protocol running underneath the IP network. The core network usually is fiber-optical in nature.[1]

[1] The division of cellular networks into the access network, backhaul, and core is a taxonomy we are introducing for discussion within this book. Different cellular protocols have different taxonomy which makes it challenging to discuss various cellular protocols into a single context.

The service network is an IP-based network and is the part of the Internet within control of the mobile network operator. Within the service network, the mobile network operator could run its various intermediary functions, for example, apply any transformations to convert video to fit into the form-factor required by a Smartphone, provide services such as email gateway interfaces. The set of services offered by any operator is very different and determined by the business needs of the operator.

Since many operators providing mobile services are multinational operators, the service network itself is a very complex IP network with many different regional and national networks. Even inside a single nation, the network operator may need to run a large network consisting of many different segments. The service network then interfaces with the rest of the Internet via one or more peering points.

A mobile application that is running on a mobile device would typically be oblivious to the specific type of cellular network it is running on. To the mobile application, the network is just another IP network. Thus, mobile applications are developed like any other application running on an IP network.

1.5 MOBILE APPLICATIONS

A mobile application has two components, one running on the mobile device and the other running on a server in the Internet. The data that mobile applications generate is exchanged between the mobile device and the server. In some cases, this data exchange maybe minimal. The application on the Smartphone side may run almost locally, with perhaps just an occasional check with the server side for any required updates.

A large number of mobile applications, however, rely upon the ability to exchange information with the servers. One big use of Smartphone is browsing the Internet for information. This application requires a constant exchange of data with servers on the Internet. Another popular use of Smartphone and tablets is watching video or music from sites providing such services. This requires an even larger amount of data flowing from the server on the Internet to the mobile device.

Mobile applications require data exchange to support a protocol between the Smartphone and the server for this data exchange. These protocols would sit above the IP layer in the hourglass shown in Figure 1.5, whereas the cellular protocols sit below the IP layer in the same hour-glass structure. In other words, the protocols that the mobile applications will use are transport and application-level protocols as described in Figure 1.2.

There are two commonly used transport protocols on IP networks, namely TCP and UDP. The TCP protocol gives the abstraction to two communicating

machines that they have a pipe of bytes between them into which bytes can be inserted at one end and extracted from the other end in the same order without any losses. The UDP protocol gives the abstraction similar to a postal delivery services where messages with addresses can be sent by one machine and received at the other machine without any assurances about reliability or ordering. Subsequently, when security became important, a protocol called Transport Layer Security (TLS) was developed to improve the security of information being exchanged. Because TCP delivery of in-order packets can have adverse impact on the real time nature of communication, several other protocols that provide interactive voice or video delivery were developed atop UDP.

The growth of the WWW gave rise to the tremendous popularity of another protocol called HTTP. HTTP operates on top of TCP and is the protocol used by the browsers most people use. The variation of HTTP which would go on TLS instead of plain TCP is Hypertext Transfer Protocol Secure (HTTPS). Another popular feature that got added to browsers would support client-side programs written in languages such as JavaScript™. A server could send some code to the browsers which could perform some of the logic and functions right at the client side.

Given the existence and popularity of the different protocols, a mobile application developer has three possible choices when developing an application on the mobile phone:

1. Write the application using HTTP protocol for data communication, with use of JavaScript (or equivalent) for local interactions.
2. Develop a private protocol for communication needs of the application which could ride directly on either TCP or UDP, depending on the needs of the applications.
3. Use a mixed set of protocols, HTTP for some of the interactions and a private protocol for the others.

The choice made by the application developers leads to three category of mobile applications: web-based mobile applications, native mobile applications, and hybrid mobile applications. The needs of the application dictate the choice among the three modes of development.

Mobile applications built on top of IP riding on top of the cellular network are the source of mobile data exchanged among people and machines. After this brief overview of the three areas, let us examine the ecosystem of this mobile data in the next chapter.