



# فصل اول

## مقدمه

بلوتوث روشنی است که به دستگاه‌ها<sup>۱</sup> امکان می‌دهد در مسافت‌های کوتاه ارتباطات بی‌سیم برقرار کنند. ارتباطات بی‌سیم از اواخر قرن ۱۹ پا به عرصه وجود نهادند و اشکال مختلفی دارند؛ از قبیل رادیو، مادون قرمز<sup>۲</sup>، تلویزیون و اخیراً ۸۰۲،۱۱. ویژگی منحصر به فرد بلوتوث نسبت به سایر روش‌های بی‌سیم، تأکید آن بر ارتباطاتی با برد کوتاه است، عموماً کوتاه‌تر از ۹،۱۴۴ متر<sup>۳</sup>. مجموعه جامعی از اسناد با عنوان راهنمای بلوتوث<sup>۴</sup>، این تکنولوژی را از سیگنال رادیویی اولیه تا سطح بالاترین پروتکل‌ها شرح می‌دهند. هدف ما کمک به شما برای یادگیری کامل مبانی و ضروریات این مبحث است تا بتوانید با استفاده از این تجربیات و مفاهیم اولیه، خودتان یادگیری را ادامه دهید؛ به عبارت دیگر هدف ما آماده کردن خواننده است.

## ۱-۱ شناخت بلوتوث از منظر برنامه‌نویسان<sup>۵</sup>

اما مطلب مشکلی در مورد بلوتوث وجود ندارد، اگرچه وسعت مطالب موجود در این زمینه آغاز یادگیری آن را اندکی دشوار می‌سازد، مطالب بسیار زیادی در سطوح گوناگون در راهنمای بلوتوث موجود است که تشخیص ضروریات را از مطالب اختیاری دشوار می‌کند. راهنمای این تکنولوژی، به ویژه مدخل‌هایی که به نظر بسیار ساده و آشنا می‌رسند (چون زیاد به گوشمان خورده‌اند)، اغلب به

1. Devices
2. Infra Red

۳. ۳۰ فوت -م-

4. <http://www.bluetooth.com/Bluetooth/Learn/Technology/Specifications/>

۵. برنامه‌نویس (Programmer) به شخصی اطلاق می‌شود که برنامه را نوشته و اشکال‌زدایی می‌کند (البته ممکن است در مراحل طراحی نیز دخالت داشته باشد اما نه همواره). درحالی‌که در کتاب از ترکیب توسعه‌دهنده نرم‌افزار (Software Developer) استفاده شده است، این ترکیب در واقع با برنامه‌نویس اندکی متفاوت است؛ چرا که در زبان انگلیسی توسعه‌دهنده نرم‌افزار به کسی گفته می‌شود که برنامه را طراحی کرده و می‌نویسد اما ممکن است برنامه‌نویس طراح یک برنامه نباشد. در این متن برای اجتناب از دراز گویی از لفظ برنامه‌نویس استفاده شده است. -م-

مطالبی خاص و بسیار تخصصی ارجاع دارند. برای مثال هیچ راهنمایی به تنهایی برای اینترنت وجود ندارد و در عوض هر یک از اجزای تشکیل‌دهنده اینترنت خود دارای راهنماهای ویژه است. برای مثال Ethernet چگونگی اتصال مجموعه‌ای از ماشین‌ها را با یکدیگر برای ایجاد یک شبکه ساده شرح می‌دهد و تنها همین مطلب در این قسمت شرح داده شده است. TCP/IP دو پروتکل خاص ارتباطی را شرح می‌دهد که پایه و اساس اینترنت را تشکیل می‌دهند، اما این بخش نیز فقط حاوی همین مفاهیم است. همین‌طور HTTP نیز پایه وب جهان‌گستر<sup>۱</sup> است که فقط به پروتکل‌های ساده‌ای خلاصه می‌شود. اینترنت مجموعه‌ای از مفاهیم ذکر شده و دیگر مفاهیم است. برنامه‌نویس در ابتدا برنامه‌نویسی اینترنت-چگونگی اتصال (برقراری ارتباط) یک کامپیوتر با کامپیوتر دیگر (در اینترنت) و ارسال و دریافت داده‌ها را یاد می‌گیرد و خود را با جزئیات Ethernet و Email درگیر نمی‌کند، چرا که هیچ‌یک از تکنولوژی‌ها قسمت اصلی را تشکیل نمی‌دهند. البته مطمئناً Email نمونه‌ای حیرت‌انگیز از برنامه‌نویسی اینترنت است و Ethernet چگونگی پیاده‌سازی ارتباطات را بیان می‌کند، اما در این مبحث برنامه‌نویسی TCP/IP مبانی کار است. از بسیاری جهات، کلمه بلوتوث مانند کلمه اینترنت است، چرا که قلمرو وسیعی از موضوعات را در بر می‌گیرد. مانند USB، Ethernet، و ۸۰۲-۱۱، بلوتوث نیز مفاهیم فیزیکی بسیاری را از قبیل فرکانس‌های (بسامدهای) رادیویی و چگونگی تعدیل و تفکیک<sup>۲</sup> سیگنال‌ها تعریف می‌کند. مانند پروتکل‌های VOIP که در بسیاری از کاربردهای اینترنت مورد استفاده قرار گرفته است، بلوتوث نیز چگونگی انتقال صدا (صوت) بین دستگاه‌ها را شرح می‌دهد و البته تمامی جزئیات این امر را نیز مشخص می‌کند. بنابراین تعجبی نیست که راهنماهای بلوتوث هزاران هزار صفحه هستند.

متن حاضر درصدد ارائه پاسخ به این سؤال است: "چگونه می‌توان برنامه‌هایی را تولید کرد که دو دستگاه بلوتوث را به یکدیگر متصل کرده و به نقل و انتقال داده‌ها بین آن دو بپردازند؟". ما مفاهیم بنیادی را به همراه روش طراحی برنامه‌ها بررسی می‌کنیم. فصل‌های آینده چگونگی پیاده‌سازی و به کار بستن این مفاهیم را در چندین زبان برنامه‌نویسی و سیستم‌عامل رایج به شما نشان خواهند داد. از همه مهمتر، سعی ما در ساده و مؤثر نگه داشتن این مطالب است.

## ۲-۱ مفاهیم اولیه برنامه‌نویسی بلوتوث

مبانی برنامه‌نویسی بلوتوث نه مشکل است و نه حجیم. در ادامه فصل، این مفاهیم با مفاهیم معادل خود در مباحث مربوط به برنامه‌نویسی اینترنت مقایسه خواهند شد. اگرچه بلوتوث از ابتدا به صورت مجزا طراحی شده است و هیچ‌گونه وابستگی به Ethernet و TCP/IP ندارد؛ راه یافتن به آن، به روشی مشابه برنامه‌نویسی اینترنت کاملاً معقول و منطقی است، چرا که تعریف عمومی برنامه‌نویسی شبکه هر دو مقوله را در بر می‌گیرد و اصول یکسانی را در برقراری ارتباط و مبادله داده‌ها بین دستگاه‌ها به

---

1. World Wide Web  
2. Demodulate

اشتراک می‌گذارد. برنامه‌نویسی اینترنت و بلوتوث نقاط مشترک بسیاری دارند، به طوری که فهم یکی از آن‌ها یادگیری دیگری را تسهیل می‌کند.

برنامه‌نویسی TCP/IP معمول و فراگیر بوده و مقایسه آن با بلوتوث، خواننده را در فهم هر دو موضوع یاری می‌کند. مهمترین تفاوت، همان‌گونه که ذکر شد، در این است که عمل بلوتوث متمرکز بر دستگاه‌هایی است که به لحاظ فیزیکی همجوار<sup>۱</sup> هستند، در حالی که در برنامه‌نویسی اینترنت فاصله فیزیکی اهمیتی ندارد. این تفاوت بر اینکه دستگاه‌ها چگونه در ابتدا یکدیگر را پیدا کرده و اولین ارتباط را برقرار می‌کنند، دارد. پس از آن، همه چیز در هر دو حیطه، تقریباً یکسان است. فرایند ایجاد ارتباط، دستگاه قصد برقراری یک ارتباط خارجی<sup>۲</sup> را داشته باشد یا پذیرش ارتباط<sup>۳</sup> ورودی را، متفاوت است. در واقع این امر تفاوت بین دستگاه گیرنده و دستگاه فرستنده است. ما اغلب این دو را متعاقباً مشتری<sup>۴</sup> و سرویس‌دهنده<sup>۵</sup> می‌نامیم.

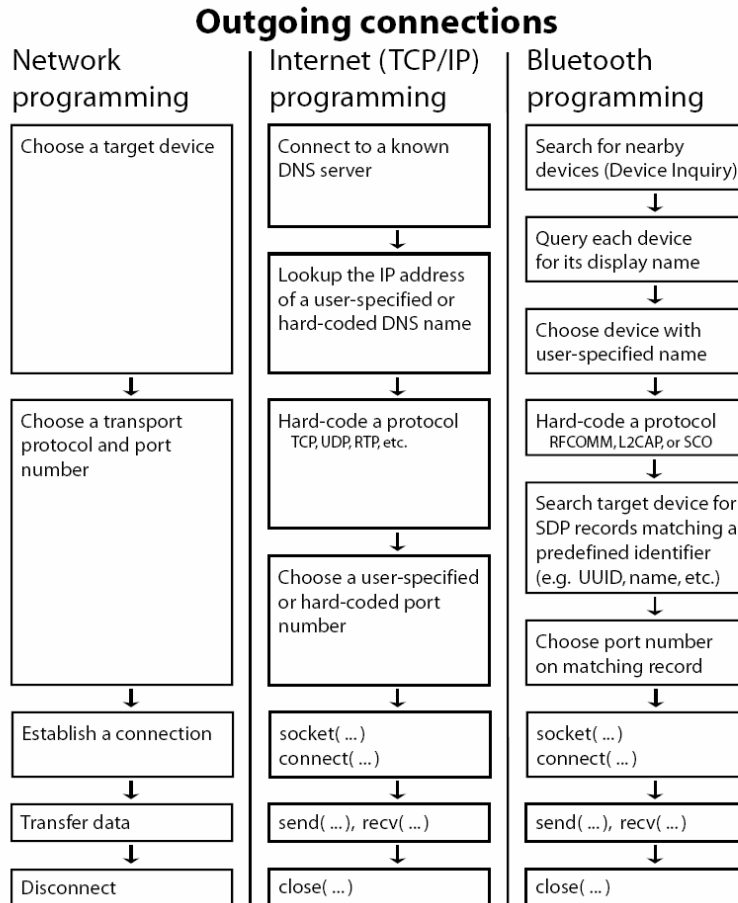
**توجه:** ما از اصطلاحات مشتری و سرویس‌دهنده، تنها برای ایجاد تمایز بین دستگاه برقرارکننده ارتباط و دستگاه پذیرنده ارتباط استفاده کرده‌ایم. این امر هیچ‌گونه ارتباطی به مدل مشتری-سرویس‌دهنده در برنامه‌نویسی شبکه ندارد. به‌رغم این همپوشانی، رفتار یک سرویس‌دهنده مانند یک مشتری و بالعکس کاملاً معقول است.

زمانی که دستگاهی یک ارتباط خارجی را برای اولین بار برقرار می‌نماید، ضروری است پیش از برقرار شدن ارتباط و انتقال داده‌ها، دستگاه مقصد و پروتکل انتقال را انتخاب کند. برای دستگاه‌هایی که ارتباطات ورودی را می‌پذیرند، انتخاب پروتکل انتقال و گوش دادن به خط پیش از پذیرش ارتباط و انتقال داده‌ها الزامی است. اشکال ۱-۱ و ۲-۱ مفاهیم اولیه و چگونگی انجام هر یک را در برنامه‌نویسی اینترنت و بلوتوث به تصویر کشیده‌اند. توجه کنید که در مورد ارتباطات خارجی فقط دو مرحله اول (انتخاب دستگاه مقصد، پروتکل انتقال و شماره درگاه) در برنامه‌نویسی بلوتوث و اینترنت با یکدیگر متفاوت است و زمانی که ارتباط خارجی برقرار شد همه چیز یکسان است.

مکانیزمی که برای پذیرش ارتباط انجام می‌گیرد در هر دو مبحث بسیار شبیه است و تفاوت اصلی، اضافه شدن امکان پشتیبانی بلوتوث از درگاه‌های تخصیص داده شده پویا است. برنامه‌نویسان مجرب شبکه متوجه این نکته خواهند شد که برخی از مراحل ذکر شده، همواره اجرا نمی‌شود و الزاماً نیازی به انجام شدن آن‌ها به ترتیبی که نمایش داده شده است، وجود ندارد. به‌عنوان مثال اگر آدرس سرویس‌دهنده‌ای در کد یک مشتری مشخص شده باشد، دیگر نیازی به استفاده از سیستم نام حوزه (DNS) یا جستجوی یک دستگاه نیست. نکته اصلی در اینجا این است که این نمودارها روال عادی و

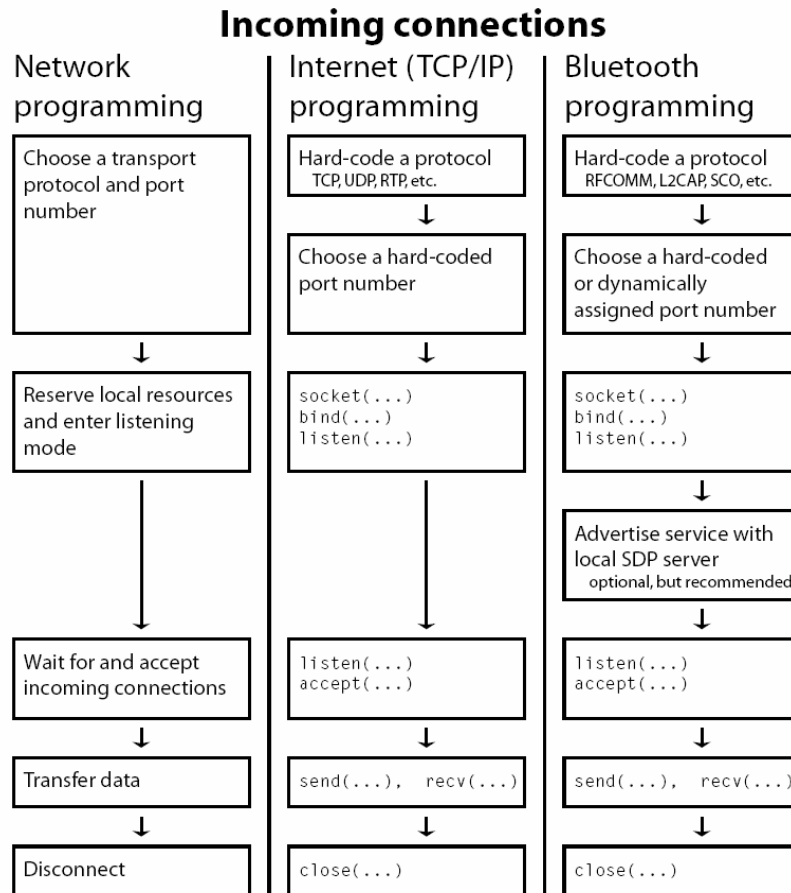
1. Physically Proximate
2. Outgoing Connection
3. Incoming Connection
4. Client
5. Server

معمول انجام این اعمال را نشان می‌دهند و می‌توان آن‌ها را در موقعیت‌های گوناگون، متناسب با آن موقعیت تغییر داد. در بخش‌های آینده با رسیدن به هر مفهوم هر یک از آن‌ها تشریح خواهد شد. فصل‌هایی که در ادامه می‌آیند چگونگی پیاده‌سازی این مفاهیم را در زبان‌های برنامه‌نویسی و سکوها<sup>۱</sup> گوناگون شرح می‌دهند.



شکل ۱-۱: به‌طور کلی پنج قدم برای ایجاد یک ارتباط خارجی وجود دارد. تنها فرایند اولیه انتخاب دستگاه مقصد، پروتکل انتقال و شماره درگاه در برنامه‌نویسی اینترنت و برنامه‌نویسی بلوتوث با یکدیگر متفاوتند.

۱. Platform معادل Environment است و به‌معنای سیستم‌عامل و یا سخت‌افزار مورد استفاده می‌باشد. -م-



شکل ۱-۲: برای پذیرش یک ارتباط نیز پنج قدم اصلی وجود دارد. مانند ارتباطات خارجی جزئیات مربوط به برنامه‌نویسی اینترنت و برنامه‌نویسی بلوتوث به یکدیگر مرتبط است، اگرچه تفاوت‌های اندکی نیز وجود دارد.

### ۱-۲-۱ انتخاب دستگاه مقصد

هر تراشه بلوتوث ساخته شده، دارای یک آدرس ۴۸ بیتی منحصر به فرد است که به آن آدرس بلوتوث یا آدرس دستگاه می‌گوییم. این آدرس در واقع چیزی شبیه به آدرس MAC<sup>۱</sup> در Ethernet است. در واقع هر دو فضای آدرس توسط سازمان یکسانی مدیریت می‌شوند<sup>۲</sup>. آدرس‌های بلوتوث در زمان ساخت مشخص می‌شوند و باید منحصر به فرد باشند و در طول عمر تراشه ثابت هستند و به صورت واحد پایه آدرس‌دهی در برنامه‌نویسی بلوتوث از آن‌ها استفاده می‌شود.

1. <http://www.ietf.org/rfc/rfc0826.txt>

2. IEEE Registration Authority

برای اینکه یک دستگاه بلوتوث بتواند با دستگاه دیگری ارتباط برقرار کند، باید آدرس بلوتوث دستگاه دیگر را به دست آورد. این آدرس در تمامی لایه‌های فرایند ارتباطی بلوتوث از پروتکل‌های رادیویی سطح پایین تا پروتکل‌های سطح بالای کاربردی مورد استفاده قرار می‌گیرد. در عوض دستگاه‌های شبکه TCP/IP که از Ethernet استفاده می‌نمایند، آدرس ۴۸ بیتی MAC را در لایه‌های بالاتر فرایند ارتباط نادیده می‌گیرند و به جای آن از آدرس‌های IP استفاده می‌کنند. با این وجود اصول، یکسان است و آدرس منحصر به فرد دستگاه مقصد برای برقراری ارتباط با آن، باید شناسایی شود.

برنامه مشتری نیاز به اطلاعات زیادی در مورد این آدرس‌های مقصد ندارد. در برنامه‌نویسی اینترنت، کاربر معمولاً نام میزبان را می‌داند، مانند `www.kernel.org` که این نام باید توسط DNS به یک آدرس IP ترجمه شود. در بلوتوث کاربر از یک نام کاربرپسند<sup>۱</sup> مانند "my phone" استفاده می‌کند و مشتری این نام را توسط جستجوی دستگاه‌های مجاور و بررسی نام هر دستگاه، به یک آدرس عددی تبدیل می‌نماید.

**توجه:** در سرتاسر کتاب از کلمهٔ آداپتور<sup>۲</sup> و دستگاه با تفاوت اندکی استفاده خواهیم کرد، در واقع یک دستگاه بلوتوث به هر ماشینی که توانایی برقراری ارتباطات بلوتوث را دارد اطلاق می‌شود. زمانی که دربارهٔ نوشتن برنامه‌ها صحبت می‌کنیم، آداپتور به دستگاه بلوتوث در کامپیوتر محلی (و یا سیستمی که برنامه را اجرا می‌کند) اطلاق می‌شود. این امر موجب کاهش ابهام می‌گردد و در مشخص کردن دستگاه مورد نظر کمک می‌کند.

### نام دستگاه

انسان‌ها توانایی برقراری ارتباط با اعداد ۴۸ بیتی مانند `0x000EED3D1829` را ندارند (همانگونه که ما نمی‌توانیم با آدرس‌های IP مانند `68.15.34.115` به درستی ارتباط برقرار کنیم) و بدین دلیل دستگاه‌های بلوتوث همواره دارای یک نام کاربرپسند هستند (به این نام، نام ظاهری<sup>۳</sup> نیز می‌گویند). این نام معمولاً به جای آدرس بلوتوث برای کاربر نمایش داده می‌شود تا دستگاه را شناسایی کند، اما در نهایت این آدرس بلوتوث است که در ارتباط واقعی مورد استفاده قرار می‌گیرد. برای بسیاری از ماشین‌ها مانند گوشی‌های تلفن همراه و کامپیوترهای رومیزی این نام قابل تغییر است و کاربر می‌تواند یک کلمه یا عبارت دلخواه را برای آن انتخاب کند و الزامی برای انتخاب یک نام منحصر به فرد وجود ندارد. البته چنین امری (عدم انتخاب نام منحصر به فرد) می‌تواند موجب سردرگمی در حالاتی که دستگاه‌های زیادی در مجاورت یکدیگر دارای نام‌های یکسان هستند، شود. برای مثال ممکن است در حالاتی نیاز به انتخاب از میان پنج گوشی مختلف وجود داشته باشد که هر یک دارای نام "my phone" هستند. نام‌ها در بلوتوث مانند نام‌های DNS هستند که در هر دوی آن‌ها اسامی، شناسه‌های کاربرپسندی هستند که به

1. User-Friendly
2. Adapter
3. Display Name

نام‌های قابل تشخیص توسط ماشین تبدیل می‌شوند. تفاوت این دو در این است که نام‌ها در DNS منحصربه‌فرد هستند (یعنی تنها می‌توان یک آدرس `www.google.com` داشت) و در یک پایگاه داده مرکزی ثبت شده‌اند، در حالی که نام‌های بلوتوث کم و بیش دلخواهی هستند و تنها در خود دستگاه ثبت شده‌اند. در TCP/IP هر موجودیت دارای یک نام DNS است. برگردان یک نام DNS به یک آدرس IP نیاز به برقراری ارتباط با یک سرویس‌دهنده نام<sup>۶</sup>، بازجست<sup>۷</sup> و انتظار برای نتیجه دارد. در بلوتوث فرایند جستجو معکوس است. در ابتدا یک دستگاه، دستگاه‌های مجاور را جستجو کرده و فهرستی حاوی آدرس‌های آن‌ها را گردآوری می‌کند. سپس با هر یک از دستگاه‌های مجاور به صورت مجزا ارتباط برقرار می‌نماید و بازجستی را برای نام کاربرپسند آن انجام می‌دهد.

### جستجوی دستگاه‌های مجاور

اکتشاف<sup>۸</sup> دستگاه، که به آن استعلام<sup>۹</sup> دستگاه نیز می‌گویند، فرایند جستجو و پیدا کردن دستگاه‌های بلوتوث مجاور است. در تئوری این کار ساده است. برای کشف دستگاه‌های مجاور یک پیام "اکتشاف" پخش می‌شود و انتظار برای دریافت پاسخ آغاز می‌گردد. هر پاسخ شامل آدرس پاسخگو و عدد صحیحی است که بیانگر رده<sup>۱۰</sup> دستگاه است (برای مثال: تلفن همراه، کامپیوتر شخصی، هدست و...). جزئیات بیشتر، مانند نام دستگاه را می‌توان پس از برقراری ارتباط با هر دستگاه به صورت مجزا به دست آورد<sup>۱۱</sup>.

در عمل، این امر موجب سردرگمی و آشفتگی برنامه‌نویسان و کاربران می‌شود. چرا که تشخیص دستگاه‌های بلوتوث مجاور ممکن است زمان طولانی را طلب کند. به عنوان مثال یک گوشی تلفن همراه و Laptop که امکان برقراری ارتباطات بلوتوث را دارند، در صورت قرار گرفتن روی یک میز معمولاً به حدود ۵ ثانیه زمان برای شناسایی یکدیگر نیاز دارند. البته در برخی موارد ممکن است این فرایند حدود ۱۰ تا ۱۵ ثانیه نیز به طول بینجامد. این زمان به نظر زیاد طولانی نمی‌آید اما غافلگیر کننده است. در طول فرایند اکتشاف گوشی تلفن همراه فرکانس‌ها را بیش از هزار بار در ثانیه تغییر می‌دهد و تنها ۷۹ فرکانس ممکن وجود دارد که می‌توان در آن‌ها به نقل و انتقال پرداخت و این امر که چرا دستگاه‌ها نمی‌توانند یکدیگر را در یک پلک به هم زدن پیدا کنند واقعاً تعجب‌آور است.

- 
1. Name Server
  2. Query
  3. Discovery
  4. Inquiry
  5. Class

۶. بلوتوث ۲.۱، روش پاسخ استعلام توسعه‌یافته را معرفی می‌کند، که اطلاعاتی که بیشتر از همه مورد درخواست قرار گرفته‌اند از قبیل نام دستگاه پاسخگو و خلاصه‌ای از سرویس‌هایی که ارائه می‌دهد را می‌توان به صورت مستقیم در پاسخ استعلام ارسال کرد، که موجب صرفه‌جویی در زمان می‌شود.

به لحاظ تکنیکی این امر به علت طراحی عجیب الگوریتم جستجو است که با جزئیات بیشتر در بخش ۱-۳-۹ توضیح داده شده است. در نسخه‌های جدید بلوتوث (از بلوتوث نسخه ۱٫۲) سعی شده تا میانگین زمان جستجو کاهش پیدا کند اما در آینده نزدیک نباید انتظار تغییر چشمگیری را داشت.

**توجه:** یکی از تصورات غلط رایج، این است که وقتی یک دستگاه بلوتوث وارد محیطی می‌شود، حضور خود را به نحوی "اعلام" می‌نماید و بدین سان دستگاه‌های دیگر از حضور آن اطلاع پیدا می‌کنند. چنین عملی هیچ‌گاه انجام نمی‌شود (اگر چه ایده خوبی است)، تنها راهی که یک دستگاه می‌تواند از حضور دیگری مطلع شود اجرای عملیات اکتشاف دستگاه است.

### جستجوپذیری<sup>۱</sup> و اتصال‌پذیری<sup>۲</sup>

به دلایل امنیتی و توان مصرفی<sup>۳</sup>، تمامی دستگاه‌های بلوتوث دارای دو گزینه اختیاری هستند که تعیین می‌کنند؛ آیا دستگاه به استعلام و تلاش برای برقراری ارتباط پاسخ بدهد یا خیر. به تفکیک، گزینه Inquiry Scan، تعیین می‌کند که آیا دستگاه به جستجو و استعلام‌ها پاسخ بدهد یا خیر و گزینه Page Scan پاسخگویی به درخواست‌های ارتباط را برعهده دارد.

هر دوی این ویژگی‌ها در جدول ۱-۱ توضیح داده شده است. این ویژگی‌ها تا حدی در اکثر دستگاه‌ها قابل پیکره‌بندی است.

جدول ۱-۱ گزینه‌های Inquiry Scan و Page Scan

تفسیر	Page Scan	Inquiry Scan
دستگاه محلی توسط سایر دستگاه‌ها قابل تشخیص بوده و درخواست‌های دستگاه‌های دیگر را برای برقراری ارتباط می‌پذیرد. در اغلب دستگاه‌ها این حالت به صورت پیش‌فرض اتخاذ شده است.	On	On
اگرچه دستگاه محلی در این حالت توسط سایر دستگاه‌ها قابل تشخیص نیست اما درخواست سایر دستگاه‌هایی که از آدرس بلوتوث آن اطلاع دارند برای برقراری ارتباط می‌پذیرد. در بعضی از دستگاه‌ها این حالت به صورت پیش‌فرض اتخاذ شده است.	On	Off
دستگاه محلی در این حالت توسط سایر دستگاه‌ها قابل تشخیص است اما درخواست سایر دستگاه‌ها را برای برقراری نمی‌پذیرد. این حالت کاربرد چندانی نداد و عموماً مورد استفاده قرار نمی‌گیرد.	Off	On
دستگاه محلی توسط سایر دستگاه‌ها قابل تشخیص نبوده و درخواست‌های سایر دستگاه‌ها را برای برقراری ارتباط نمی‌پذیرد. این حالت عموماً در مواقعی که دستگاه محلی فقط ارتباطات خارجی را برقرار می‌کند مفید است.	Off	Off

1. Discoverability
2. Connectability
3. Power

اگرچه گزینه‌های Inquiry Scan و Page Scan گنج‌کننده هستند، ولی این دو نباید با فرایندهای اصلی تشخیص و برقراری ارتباط با دیگر دستگاه‌ها اشتباه گرفته شود. دلیل چنین نام‌گذاری در این است که این گزینه‌ها تعیین می‌کنند که آیا دستگاه محلی، جستجو و تلاش دستگاه‌های دیگر را برای برقراری ارتباط، بررسی<sup>۱</sup> می‌نماید یا خیر. یک دستگاه در صورتی که گزینه Inquiry Scan آن فعال باشد، قابل جستجو است.

### ۱-۲-۲ انتخاب پروتکل انتقال

برنامه‌های کاربردی گوناگون دارای نیازهای متنوعی هستند و به همین دلیل پروتکل‌های انتقال متفاوتی نیز وجود دارد. این بخش پروتکل‌هایی را که برای برنامه‌نویسی بلوتوث باید از آن‌ها مطلع باشید، به همراه کاربرد آن‌ها در برنامه‌های گوناگون شرح می‌دهد.

دو عامل مهم که موجب ایجاد تفاوت بین پروتکل‌های مورد بحث می‌شود، تعهد<sup>۲</sup> و معنا<sup>۳</sup> است. منظور از تعهد پروتکل، میزان تلاشی است که پروتکل برای تحویل بسته<sup>۴</sup> ارسال شده توسط برنامه به کار می‌بندد. یک پروتکل قابل اعتماد، مانند TCP، دارای نگرش "تحویل-یا-مرگ" است، این پروتکل، یا از تحویل تمامی بسته‌های ارسال شده اطمینان حاصل می‌نماید و یا ارتباط را قطع می‌نماید. یک پروتکل "کوشا"<sup>۵</sup> مانند UDP تلاشی منطقی را برای تحویل بسته‌های ارسال شده انجام می‌دهد، اما شرایطی را که ممکن است بسته‌ها به هر دلیلی مانند امواج مٌخل، خرابی مسیریاب و ... به مقصد نرسند نادیده می‌گیرد.

معنای پروتکل (روش انتقال بسته) می‌تواند مبتنی بر بسته<sup>۶</sup> یا مبتنی بر جریان<sup>۷</sup> باشد. یک پروتکل مبتنی بر بسته مانند UDP داده‌ها را در داده‌گرام‌های<sup>۸</sup> مجزا با یک طول مشخص و محدود ارسال می‌نماید. یک پروتکل مبتنی بر جریان، مانند TCP، داده‌ها را بدون نگرانی درباره نقطه پایان یک بسته و آغاز دیگری ارسال می‌کند.

بلوتوث، پروتکل‌های انتقال زیادی دارد که هر یک کاربرد خاص خود را دارند. ما تنها چهار پروتکل ضروری را بررسی می‌کنیم. البته برای شروع، تنها دو پروتکل RFCOMM و L2CAP کافی است و از میان این دو نیز، RFCOMM، برای بسیاری از خواننده‌ها کفایت می‌کند. این پروتکل‌ها براساس رابطه (وابستگی به یکدیگر) ارائه شده‌اند. البته خواننده مشتاق می‌تواند مروری بر چند پروتکل دیگر نیز داشته باشد.

1. Scan
2. Guarantee
3. Semantics
4. Best-Effort
5. Packet Based
6. Stream Based

۷. Datagram : به یک موجودیت جامع، و غیر وابسته داده‌ای که اطلاعات لازم را برای مسیریابی شدن از مبدأ به مقصد در شبکه در خود جای می‌دهد و به بسته‌های قبلی مبادله شده بین مبدأ و مقصد و شبکه موجود اتکا ندارد داده گرام می‌گویند. -م-

**RFCOMM**

پروتکل RFCOMM<sup>۱</sup>، قابل اعتماد و مبتنی بر جریان است. قابلیت اعتماد و خدمات این پروتکل تقریباً همسطح TCP است. در راهنمای بلوتوث ذکر شده که این پروتکل برای شبیه‌سازی درگاه‌های سری RS-232 طراحی شده است. (تا اضافه کردن قابلیت‌های بلوتوث به دستگاه‌هایی که دارای درگاه‌های سری هستند، توسط سازندگان تسهیل شود). اما ما ترجیح می‌دهیم که این تعریف را کنار گذاشته و بگوییم که RFCOMM یک پروتکل انتقال چند منظوره است که می‌توان از آن برای شبیه‌سازی درگاه‌های سری هم استفاده کرد.

از دیدگاه برنامه‌نویس شبکه، انتخاب شماره درگاه<sup>۲</sup> مهمترین تفاوت بین TCP و RFCOMM است. چرا که TCP از ۶۵۵۳۵ درگاه باز روی یک ماشین پشتیبانی می‌نماید، اما این عدد در RFCOMM تنها به ۳۰ درگاه محدود می‌شود. این امر تأثیر بسزایی بر چگونگی انتخاب درگاه برای سرویس‌دهنده‌ها دارد. یکی از وجوه تمایز RFCOMM، بسته به کاربرد و سکوی مقصد، در این است که گاهی اوقات تنها انتخاب ممکن است. برخی از محیط‌ها مانند Microsoft Windows XP Bluetooth API و Nokia Series 60 Python تنها از پروتکل انتقال RFCOMM پشتیبانی می‌کنند. البته این موضوع چندان هم بد و محدود کننده نیست، چرا که RFCOMM یک پروتکل قدرتمند و چند منظوره است. البته به یاد داشتن سایر پروتکل‌ها نیز خالی از لطف نیست، برای اطلاعات بیشتر بخش ۱-۳-۱۰ را مشاهده نمایید.

**L2CAP**

L2CAP<sup>۳</sup> پروتکلی مبتنی بر بسته است که قابلیت اعتماد آن را می‌توان برای سطوح مختلفی تنظیم کرد<sup>۴</sup>. حداکثر اندازه بسته در حالت پیش‌فرض ۶۷۲ بایت است که می‌توان پس از برقرار شدن ارتباط این اندازه را تا ۶۵۵۳۵ بایت نیز افزایش داد. L2CAP را می‌توان با UDP که یک پروتکل مبتنی بر بسته کوشا است، مقایسه کرد. البته تفاوت‌هایی نیز وجود دارد که موجب می‌شود حوزه کاربرد L2CAP گسترده‌تر از UDP باشد. هر دوی آن‌ها پروتکل‌های مبتنی بر بسته هستند، اما در L2CAP رعایت ترتیب در تحویل بسته‌ها اجباری است. در UDP زمانی که یک کامپیوتر دو بسته را ارسال می‌کند، ممکن است بسته دوم به دلیل اتفاقاتی که در حین مسیریابی اینترنتی احتمال وقوع دارد، قبل از اولی به گیرنده برسد. چنین حالتی هیچ‌گاه در L2CAP اتفاق نمی‌افتد و بسته‌ها به همان ترتیب ارسالشان به گیرنده می‌رسند. به‌علاوه UDP قابلیت اعتماد ثابت و غیرقابل تغییری دارد، درحالی‌که قابلیت اعتماد در L2CAP را می‌توان در سطوح گوناگونی تنظیم کرد. این تفاوت‌ها بدین معنی هستند که L2CAP را می‌توان در

---

1. Radio Frequency Communication

2. Port Number

3. Logical Link Control and Adaption Protocol

۴. این امر برای تمامی مقاصد و اهداف دقیق است اما از لحاظ تکنیکی صحیح نیست. برای جزئیات بیشتر بخش بعد (ACL) را ببینید.