

مهندسی اینترنت

مقدمه‌ای بر لایه کاربرد

الیاس محمدزاده کوثری

<http://elyas.kosari.fumblog.um.ac.ir>

Email: elyas.kosari@gmail.com

الگوی مشتری/خدمت گذار

- هدف یک شبکه فراهم کردن سرویس‌هایی برای کاربران است.
- کاربری در سایت محلی، خواستار دریافت سرویسی از کامپیوتری در سایت راه دور است.
- یک راه برای رسیدن به این هدف، استفاده از دو برنامه است.
- کامپیوتر محلی برنامه‌ای را اجرا می‌کند تا سرویسی را از کامپیوتر راه دور درخواست کند؛
- کامپیوتر راه دور برنامه‌ای را اجرا می‌کند که تا سرویسی را به برنامه درخواست کننده ارائه دهد.
- بدین معنی که دو کامپیوتری که توسط اینترنت به هم متصل شده‌اند، هر یک باید برنامه‌ای را اجرا کنند: یکی برای فراهم کردن سرویس و دیگری برای درخواست سرویس.

الگوی مشتری/خدمت گذار ...

- برای پیاده‌سازی این دستاورد، باید به سؤالات زیر پاسخ داد:
 - آیا هر دو کاربرد، باید توانایی درخواست سرویس و فراهم کردن آن را داشته باشند؛ یا هر یک باید فقط یکی از دو کار را انجام دهند؟
 - استفاده از دو برنامه: مشتری (Client) و خدمت‌گذار (Server)
 - آیا خدمت‌گذار باید سرویس خود را فقط برای یک مشتری خاص فراهم کند یا برای تمامی مشتری‌هایی که خواستار آن سرویس هستند؟
 - سرویس معمولاً برای تمامی مشتری‌هایی که درخواست آن را می‌دهند، فراهم می‌شود.
 - آیا هر کامپیوتر باید فقط یکی از این دو برنامه را اجرا کند؟
 - هر کامپیوتری که به اینترنت وصل می‌شود باید توانایی اجرای هر برنامه مشتری را داشته باشد؛ ولی معمولاً برای اجرای برنامه خدمت‌گذار باید از کامپیوتری استفاده شود که بتواند به طور پیوسته سرویس‌دهی کند.

الگوی مشتری/خدمت‌گذار ...

- یک برنامه باید دائماً اجرا شود یا فقط زمانی که سرویسی وجود دارد؟
 - معمولاً برنامه مشتری فقط زمانی اجرا می‌شود که درخواست سرویسی وجود داشته باشد؛ ولی برنامه خدمت‌گذار باید به طور دائمی اجرا شود.
- باید یک برنامه جهانی خدمت‌گذار وجود داشته باشد که همه سرویس‌ها را فراهم می‌کند، یا برای هر سرویسی باید برنامه خدمت‌گذار مجزایی داشته باشیم؟
 - معمولاً برای هر سرویسی، برنامه خاص خود وجود دارد.
- در ادامه به بررسی بخش‌های مختلف چنین سیستمی می‌پردازیم.

خدمت گذار

- خدمت گذار (Server) برنامه‌ای است که بر روی یک ماشین راه دور اجرا می‌شود و سرویسی را برای مشتری‌ها فراهم می‌کند.
- با اجرای این برنامه، منتظر دریافت درخواست‌های مشتری‌ها می‌گردد تا به آنها سرویس دهد ولی هیچگاه خودش سرویسی را بدون درخواست آغاز نمی‌کند.
- این برنامه یک برنامه بی‌نهایت است؛ بدین معنی که به طور بی‌نهایت باید در حال اجرا باشد.

مشتری

- مشتری (Client) برنامه‌ای است که بر روی یک ماشین محلی اجرا می‌شود و درخواست سرویسی از خدمت‌گذار را دارد.
- برنامه مشتری، برنامه‌ای محدود است؛ بدین معنی که کاربر آن را اجرا می‌کند و با پایان یافتن سرویس، آن را خاتمه می‌دهد.
- معمولاً مشتری با استفاده از IP میزبان راه دور و نیز پورت شناخته شده مربوط به یک برنامه خدمت‌گذار خاص، یک کانال ارتباطی را باز می‌کند.
- پس از باز شدن کانال ارتباطی، مشتری درخواست خود را ارسال کرده و پاسخ را دریافت می‌کند.

همزمانی

- مشتری و خدمت‌گذار، هر دو می‌توانند به طور همزمان اجرا گردند؛ بدین معنی که می‌توان چندین مشتری یا چندین خدمت‌گذار در حال اجرا به طور همزمان داشت.

همزمانی در مشتری‌ها

- مشتری‌ها در یک ماشین می‌توانند به صورت تکراری یا به طور همزمان اجرا گردند.
- اجرای مشتری‌ها به صورت تکراری به معنی اجرای آنها یکی پس از دیگری است؛ یک مشتری باید اجرا گردد و خاتمه یابد، قبل از اینکه ماشین بتواند مشتری بعدی را آغاز کند.
- با این حال، امروزه بیشتر کامپیوترها اجازه اجرای مشتری‌ها به طور همزمان را می‌دهند؛ بدین معنی که دو یا بیشتر برنامه مشتری می‌توانند همزمان اجرا گردند.

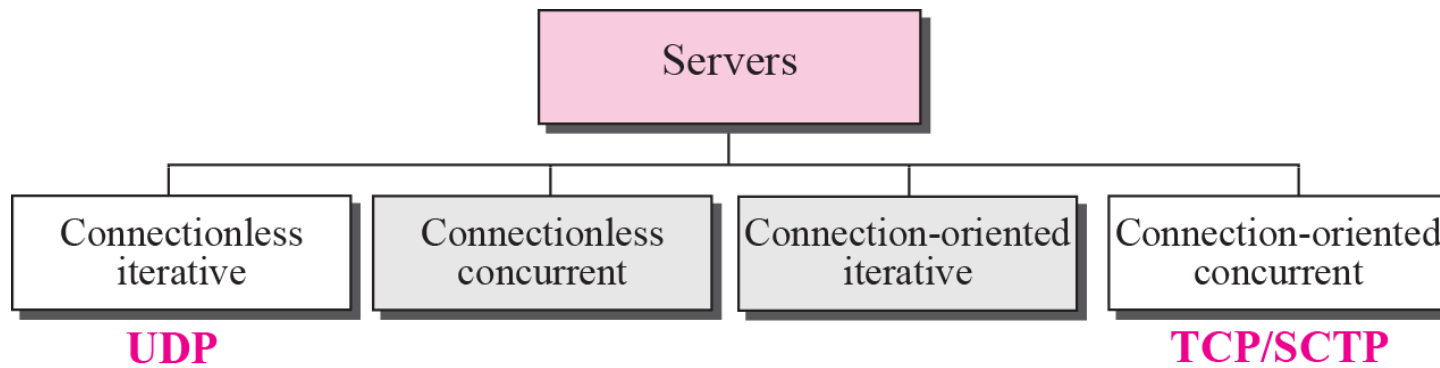
همزمانی در خدمت‌گذارها

- خدمت‌گذار تکراری، تنها به یک درخواست در آن واحد می‌تواند پاسخ دهد.
- چنین خدمت‌گذاری، درخواستی را دریافت کرده، آن را پردازش می‌کند و پاسخ مربوط به آن را ارسال می‌کند، قبل از اینکه بتواند به درخواست بعدی پاسخ دهد.
- از طرف دیگر خدمت‌گذار همزمان، می‌تواند درخواست‌های بسیاری را به طور همزمان پردازش کند و بنابراین زمان خود را بین چندین درخواست به اشتراک بگذارد.

همزمانی در خدمت‌گذارها ...

- خدمت‌گذارها یا از UDP استفاده می‌کنند و یا از TCP/SCTP.
- UDP پروتکلی بدون اتصال است و TCP/SCTP هر دو پروتکل‌های اتصال‌گرا هستند.
- بنابراین عملیات خدمت‌گذار به دو عامل بستگی دارد: پروتکل لایه انتقال و نحوه ارائه سرویس.
- به لحاظ تئوری می‌توان چهار نوع خدمت‌گذار مختلف داشت (اسلاید بعد).

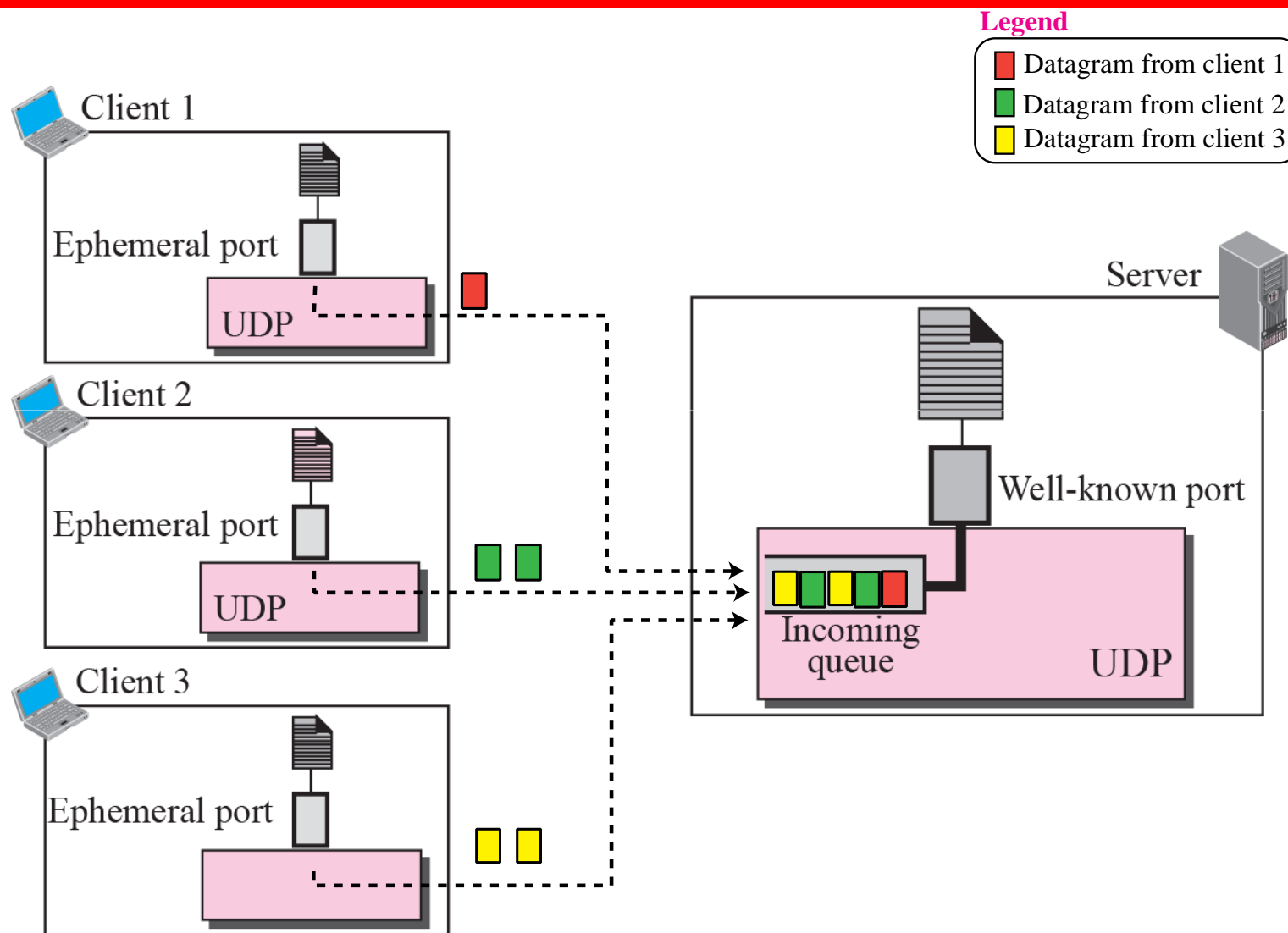
انواع خدمت گذارها



خدمت‌گذار پشت سر هم بدون اتصال

- خدمت‌گذارانی که از UDP استفاده می‌کنند معمولاً تکراری هستند، بدین معنی که به درخواست‌ها یکی پس از دیگری پاسخ می‌دهند.
- خدمت‌گذار درخواست دریافتی در داده‌گرام UDP را می‌گیرد، آن را پردازش می‌کند، و پاسخ را به UDP تحویل می‌دهد تا به شمتری ارسال گردد.
- خدمت‌گذار توجهی به سایر داده‌گرام‌ها نمی‌کند.
- این داده‌گرام‌ها در صفی ذخیره می‌شوند و منتظر خدمت‌رسانی می‌شوند.
- این داده‌گرام‌ها می‌توانند همگی از یک مشتری باشند یا از مشتری‌های مختلفی آمده باشند.
- در هر دو صورت، یکی پس از دیگری پردازش می‌شوند.

خدمت گذار پشت سر هم بدون اتصال ...



خدمت‌گذار همزمان اتصال‌گرا

- خدمت‌گذارانی که از TCP (یا SCTP) استفاده می‌کنند معمولاً همزمان هستند.
- چنین خدمت‌گذاری می‌تواند به مشتری‌های بسیاری به طور همزمان سرویس دهد.
- ارتباط اتصال‌گرا است؛ یعنی درخواست به صورت دنباله‌ای از بایت‌ها است که می‌تواند در چندین سگمنت دریافت گردد.
- بین مشتری و خدمت‌گذار اتصالی برقرار می‌شود، و تا پایان ارسال داده‌ها باز می‌ماند.

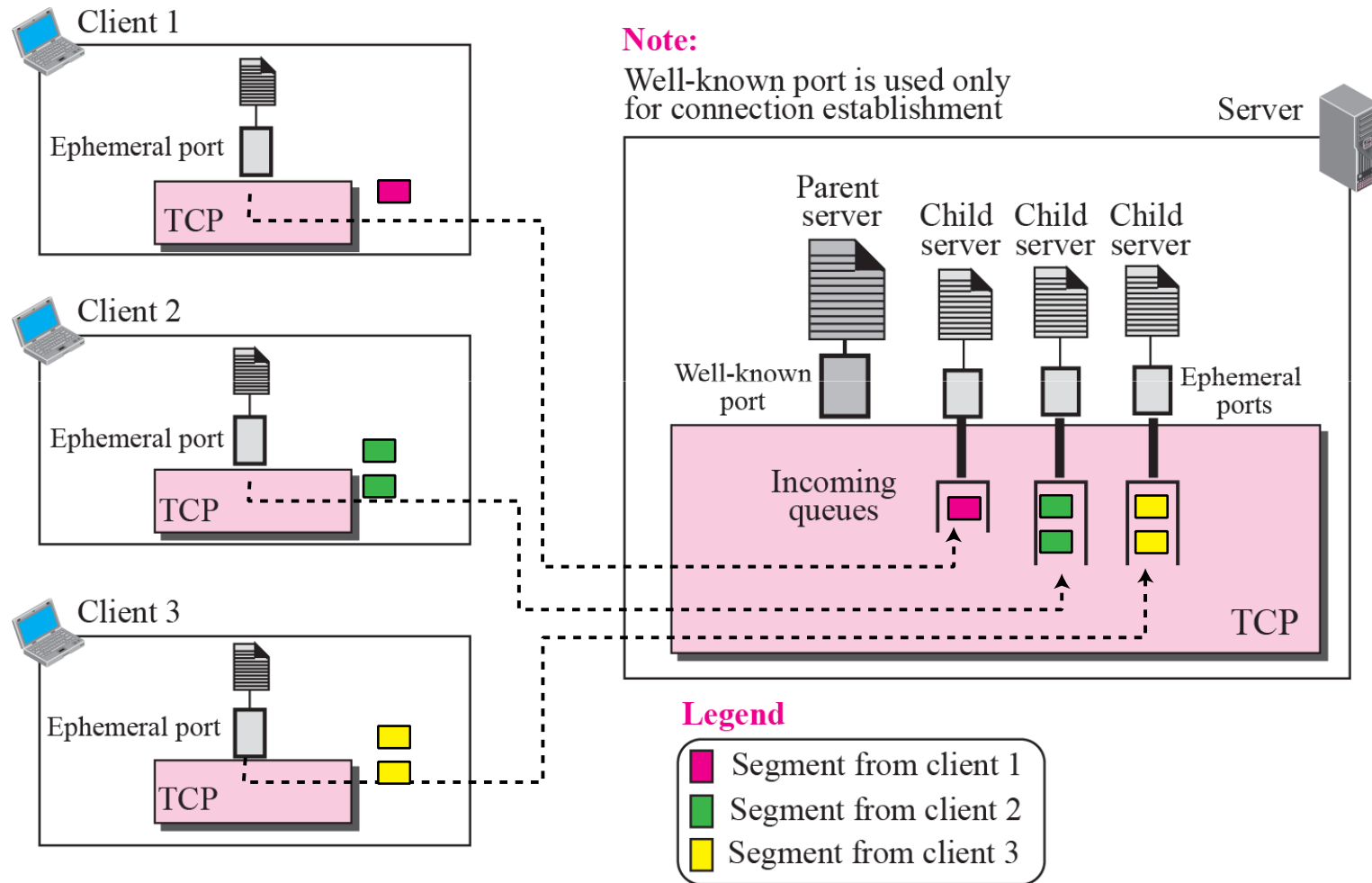
خدمت‌گذار همزمان اتصال‌گرا ...

- چنین خدمت‌گذاری نمی‌تواند فقط از یک پورت استفاده کند زیرا هر اتصال نیاز به یک پورت دارد و ممکن است اتصال‌های بسیاری به طور همزمان باز باشند.
- بنابراین به پورت‌های بسیاری نیاز داریم، ولی هر خدمت‌گذار تنها می‌تواند از یک پورت شناخته شده استفاده کند.
- راه حل این است که یک پورت شناخته شده داشته باشیم و سایر پورت‌ها، موقتی باشند.
- خدمت‌گذار اتصال را در پورت شناخته شده می‌پذیرد. مشتری برای برقراری اتصال اولیه با این پورت ارتباط برقرار می‌کند.

خدمت‌گذار همزمان اتصال‌گرا ...

- پس از برقراری اتصال، خدمت‌گذار یک پورت موقتی را به این اتصال نسبت می‌دهد تا پورت شناخته شده را آزاد کند.
- حال تبادل داده می‌تواند بین این دو پورت موقت انجام شود.
- برای سرویس‌دهی به چندین مشتری، خدمت‌گذار فرآیندهای فرزند (Child) را ایجاد می‌کند که کپی فرآیند اصلی هستند.

خدمت گذار همزمان اتصال گرا ...



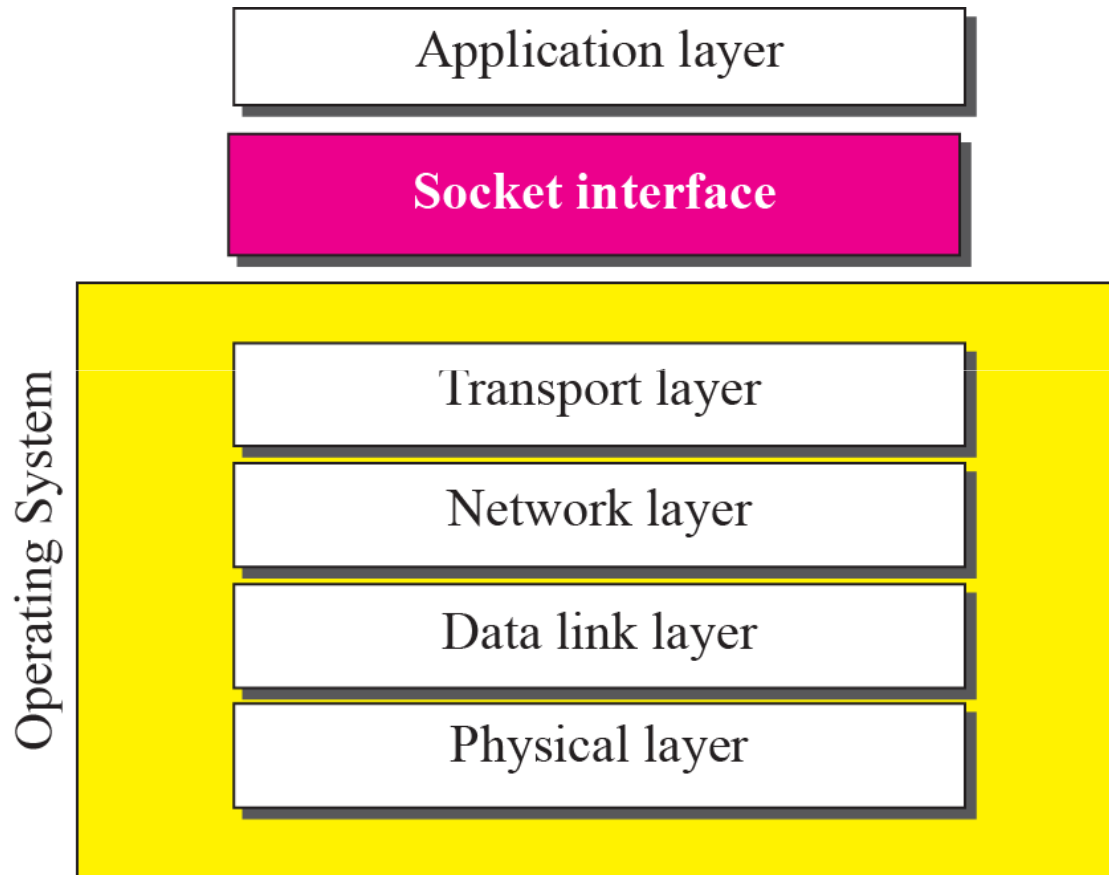
واسط سوکت

- یک برنامه کامپیوتری، مجموعه‌ای از دستورات است که کاری را کامپیوتر باید انجام دهد را مشخص می‌کند.
- اگر نیاز به برنامه‌ای داشته باشیم که قادر به برقراری ارتباط با برنامه دیگری در کامپیوتر دیگری باشد، به مجموعه دستوراتی نیاز داریم که از لایه انتقال بخواهند که اتصالی را باز کند، داده‌هایی را به/از کامپیوتر دیگر ارسال/دریافت کند، و اتصال را ببندند.
- به چنین مجموعه دستوراتی **واسط** گفته می‌شود.
- به عبارتی **واسط مجموعه دستوراتی است که برای تعامل بین دو موجودیت طراحی شده است.**

واسط سوکت ...

- چندین نوع واسط مختلف طراحی شده‌اند، که سه تای آنها راجی‌تر هستند:
 - واسط سوکت (Socket Interface)
 - واسط لایه انتقال (Transport Layer Interface یا TLI)
 - دنباله (STREAM)
- در این بخش تنها به واسط سوکت می‌پردازیم.
- واسط سوکت در اوایل دهه ۱۹۸۰ به عنوان بخشی از محیط UNIX طراحی گردید.
- برای فهم بهتر واسط سوکت، باید رابطه بین سیستم عامل و مجموعه پروتکل TCP/IP را بدانیم.

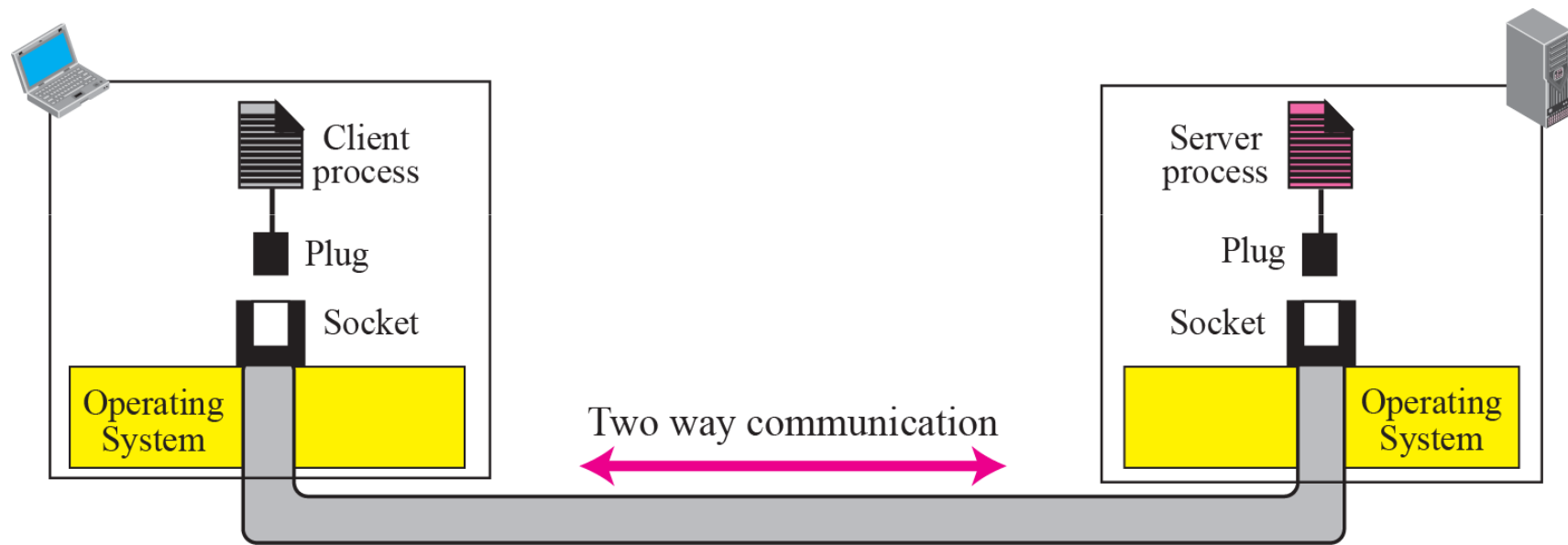
واسط سوکت ...



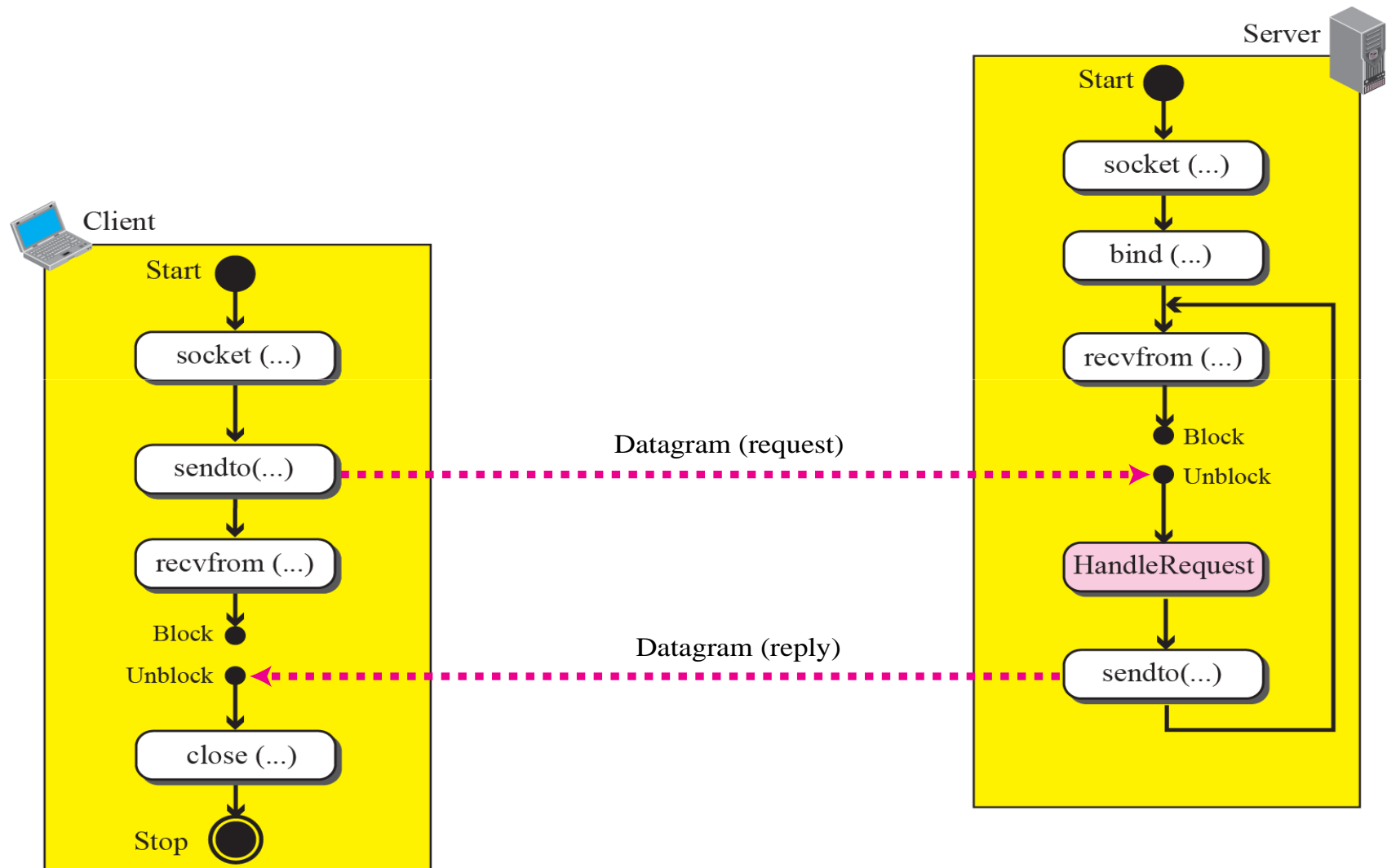
سوکت

- **سوکت** نرم‌افزاری انتزاعی است که یک سوکت سخت‌افزاری را شبیه‌سازی می‌کند.
- برای استفاده از کانال ارتباطی، برنامه کاربردی (مشتری یا خدمت‌گذار) باید از سیستم عامل درخواست کند تا یک سوکت ایجاد کند.
- سپس برنامه کاربردی می‌تواند به سوکت **متصل** شود تا بتواند داده ارسال/دریافت کند.
- برای انتقال داده، به یک جفت سوکت نیاز داریم، یکی در هر یک از دو برنامه کاربردی.
- در اینترنت، سوکت یک ساختمان داده نرم‌افزاری است که در ادامه شرح می‌دهیم.

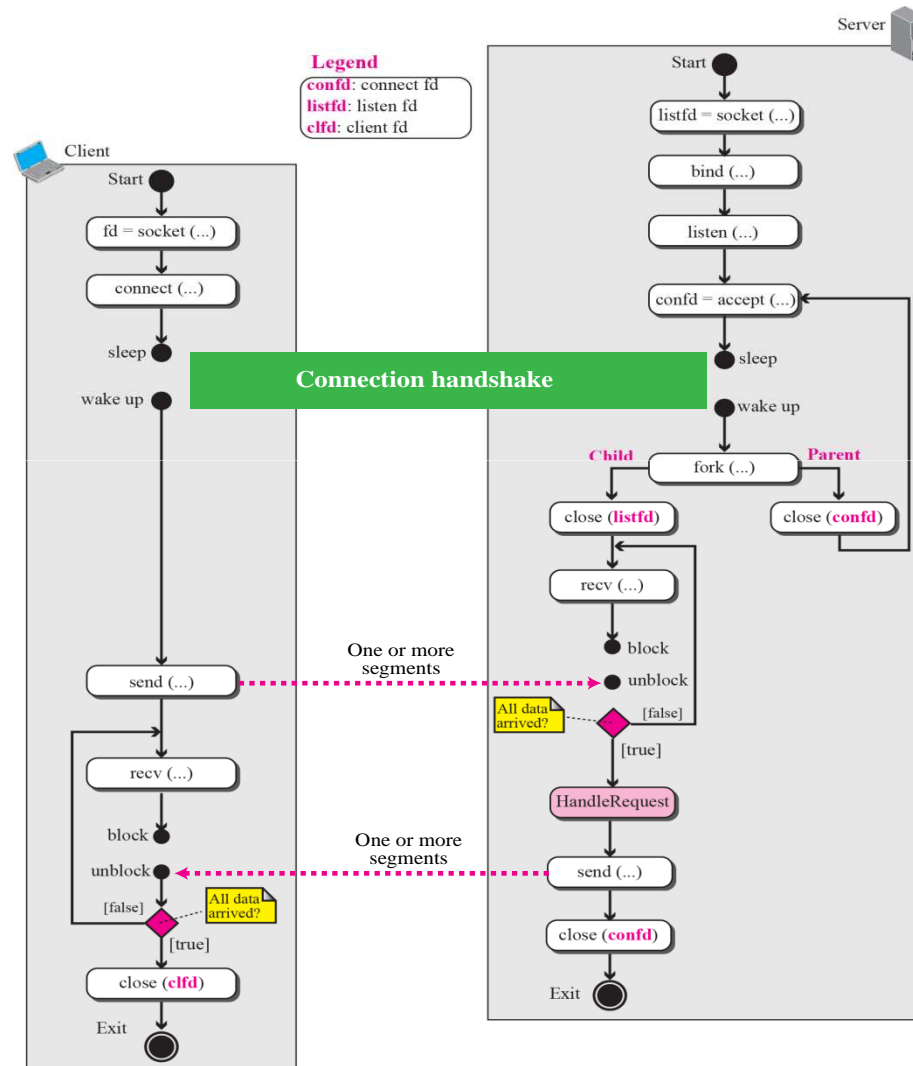
مفهوم سوکت



ارتباط با استفاده از UDP

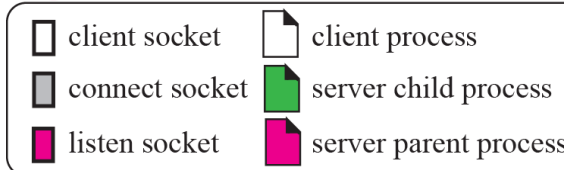


ارتباط با استفاده از TCP



وضعیت فرآیندهای پدر و فرزند با توجه به سوکت

Legend



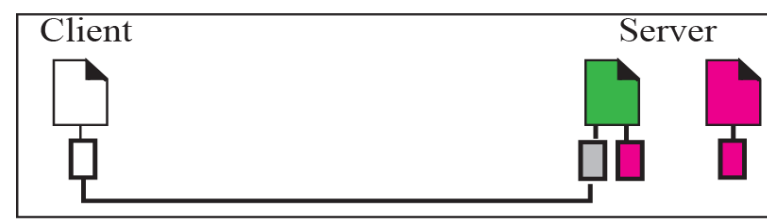
a. Before return from accept



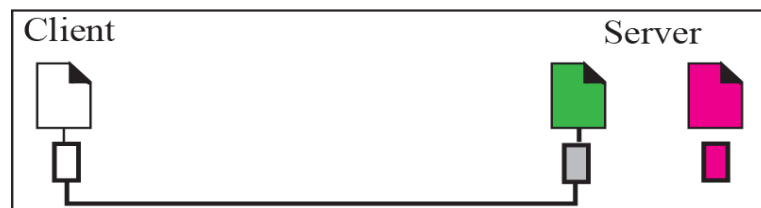
b. After return from accept



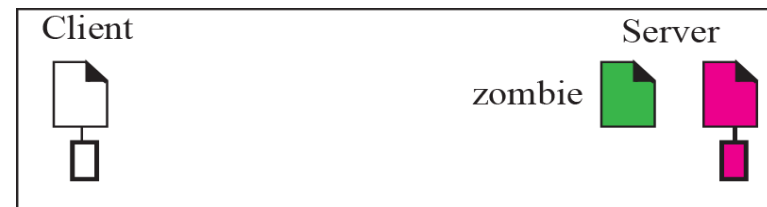
c. After fork



d. After parent closes connect socket



e. After child closes listen socket



f. After child closes connect socket