لایه انتقال

لایه انتقال از سرویسهای مهیا شده توسط لایه شبکه مانند انتخاب بهترین مسیر و آدرس دهی منطقی استفاده نموده و ارتباطی End-to-End بین مبدأ و مقصد ایجاد می کند. این لایه متشکل از یک سلسله مراتب قرارداد است که قابل اعتماد و مقرون به صرفه داده ها از ماشین منبع به ماشین مقصد و به صورت مستقل از شبکه فیزیکی را بر عهده دارند. این لایه با فراهم آوردن خدمات سازماندهی شده و مطمئن برای برنامه های کاربردی در لایه های بالاترؤ مشکلات و ناکارآمدی لایه IP را نیز جبران می نماید. کاستی های مربوط به لایه IP که بایستی در لایه انتقال پوشانده شوند عبارتند از:

۱- پس از ارسال متوالی بسته های داده، در مقصد ممکن است به صورت نامرتب و خارج از ترتیب ارسالشان دریافت شوند.

۲- ممکن است پس از ارسال یک بسته، به علت تأخیر در دریافت آن در سمت گیرنده، بسته مجدداً توسط فرستنده ارسال گردد که این امر توسط لایه IP قابل تشخیص نیست و در هر بسته در سمت گیرنده به ماشین تحویل می گردد. بنابراین بسته دوم بایستی توسط لایه انتقال تشخیص و حذف گردد.

۳- برروی هر یک از ماشینهای واقع در شبکه می تواند چند برنامه به صورت همزمان یا چند کاربره در حال اجرا باشد و هر کدام از آنها نیز نیازمند ارسال و دریافت بسته های داده برروی شبکه باشند. در لایه IP تمایزی بین بسته های ارسالی دو برنامه در حال اجرا برروی یک ماشین وجود ندارد. بنابراین در لایه شبکه بایستی بتوان هر بسته را به پردازهٔ مربوط به خود انتساب داد.

٤- با توجه به امکان متفاوت بودن توان تولید بسته های ارسالی در مبدأ و پردازش آنها در مقصد، لذا بایستی برروی سرعت تحویل بسته ها به یک ماشین کنترل صورت بگیرد که این کنترل در لایه IP فراهم نیست لذا در لایه انتقال بایستی تضمین نمود که ماشین گیرنده بسته های ارسالی به علت توان پردازش و استفاده از بسته های ارسالی و برداشتن آنها از روی شبکه، هیچ بسته ارسالی مربوط به خود را از دست ندهد.

بر اساس موارد مطرح شده در بالا، توابع ارائه شده در لایه انتقال عبارتند از:

- تقسیم بندی داده های برنامه های کاربردی لایه های بالاتر: لایه انتقال توانایی گنجاندن داده های مربوط به برنامه های کاربردی

مختلف در یک جریان داده متعلق به ماشین را دارد. این امر توسط اختصاص مشخصه ای به نام شماره پورت برای هر برنامه کاربردی که به صورت همزمان در حال اجرا با برنامه های دیگر برروی یک ماشین است محقق می گردد که شرح آن در ادامه بحث ذکر خواهد گردید.

- ایجاد ارتباط End-to-End بین فرستنده و گیرنده.

بدین مفهوم که دریافت یک و تنها یک کپی از بسته ارسالی در سمت گیرنده و همچنین مرتب سازی باسته های دریافتی بر اساس ترتیب ارسال آنها توسط این تابع فراهم گردد.

- مهیا نمودن سرویسهای انتقال از ماشین مبدأ به ماشین مقصد.
- کنترل جریان داده و تضمین قابلیت اعتماد داده: لایه انتقال جامعیت داده دریافتی را بوسیله کنترل جریان داده و قادر نمودن کاربر به درخواست انتقال داده بین دو ماشین مبدأ و مقصد، تأمین می نماید. کنترل جریان داده مکانیزمی است که به میزبانهای انتقال اجازه می دهد تا برروی حجم داده های ارسال در هر زمان توافق نمایند. برای بدست اوردن آنتقال داده قاغبل

اعتماد نیز از یک ارتباط اتصال گرا جهت ایجاد ارتباط بین فرستنده و گیرنده استفاده می گردد.

کنترل جریان داده: زمانی که لایه انتقال بسته های داده خود را ارسال می نمایند می تواند جامعیت داده ارسالی را نیز تضمین کند که یکی از روشهای آن کنترل جریان نامیده می شود. کنترل جریان از ارسال داده در سمت گیرنده به گونه ای که بافرهای موجود در گیرنده را سرریز نمایدت، جلوگیری می نماید. سرریز باعث می گردد که قسمتی از داده در سمت گیرنده از دست برود که این امر در انتقال داده بسیار خطرناک می نماید.

آدرس پورت:

همانگونه که ذکر گردید برروی هر ماشین موجود در شبکه می تواند پروسه های همزمان در حال اجرا باشند و هرکدام از آنها بخواهد از امکانات شبکه استفاده نماید. هر پروسه برای تقاضای برقراری ارتباط با پروسه دیگر برروی شبکه، یک شماره شناساسیی برای خود برمی گزیند که این شماره شناسایی تنها در لایه انتقال مفهوم داشته و در لایه IP قابل تشخیص نیست. این شماره شناسایی را

آدرس پورت می نمایند. در سرآیند بسته ای که توسط پروتکل انتقال سازماندهی می شود، آدرس پورت فرستنده و آدرس پورت پروسه گیرنده آن درج می گردد. یکتا بودن شماره های پورت که به پروسه ها رسمیت و هویت یکتا می بخشد، توسط پروتکل کنترل انتقال به عنوان جزئی از سیستم عامل نظارت خواهد شد. بدین صورت که سیستم عامل جدولی را نگهداری می کند که شماره شناسایی تقاضا دهنده ارتباط در آن وجود دارد. آدرس پورت شماره ای دوبایتی است که بدین لحاظ می تواند عددی بین صفر تا ٦٥٦٣٥ باشد باشد که این نشاندهنده این است که از نظر لایه انتقال حدود ٦٥ هزار پروسه به صورت همزمان می توانند برروی یکی از ماشینهای فرستنده یا گیرنده از امکانات شبکه استفاده نمود و یا پروسه های دیگر موجود در میزبانهای دیگر ارتباط برقرار نمایند (ابده آل). شماره پورتهای کمتر از ٢٥٦، پورتهای معروف نام دارند و برای خدمات استاندارد رزرو گردیده اند. به عنوان مثال هر فرآیندی که مایل به برقراری اتصال با میزبانی باشد تا فایل را به کمک FTP انتقال دهد، می تواند به پورت ٢١ میزبان مقصد وصل شود و یا پورت ٢٣ که برپا ارتباط راه دور با سرور (Telnet) مورد استفاده

قرار می گیرد. ترکیب آدرس پورت و آدرس IP را که نشاندهنده یک پروسه خاص در یک ماشین خاص در شبکه است را سوکت می نامیم. در هنگام ایجاد ارتباط در سمت گیرنده و نیز در سمت فرستنده، یک سوکت ایجاد می گردد. در سوکت ایجاد شده در سمت مشتری، آدرس پورت به صورت دلخواه و متغیر انتخاب می گردد و در سمت سرور سوکت ایجاد دشه حاوی آدرس مشخص و شناخته شده ای برای مشتریهای آن برنامه در حال اجرا است.

در لایه انتقال درپروتکل به نامهای TCP و UDP تعریف شده اند که ابتدا پروتکل TCP و سپس UDP را معرفی خواهیم کرد.

پروتکل کنترل Transmission Control Protcol (TCP)

TCP پروتکی اتصالگر و قابل اعتماد است. در یک محیط اتصالگرا، یک ارتباط بین فرستنده و گیرنده قبل از انتقال اطلاعات بین آنها، ایجاد می نماید. TCP مسئول شکستن پیامها به قطعات و سپس بازسازی پیام توسط اتصال قطعات در مقصد و نیز ارسال مجدد پیامهایی که در بین راه به هر علتی از بین می روند می باشد. در

حقیقت TCP، مداری مجازی بین برنامه های کاربردی نهایی ایجاد می نماید.

به پروتکلهایی که قبل از مبادله داده ها سعی در برقراری یک ارتباط و ایجاد هماهنگی قبلی می نمایند، پروتکلهای اتصالگرا گویند. همانگونه قبلاً ذکر شد یکی از کاستیهای لایه IP عدم تضمین در آماده بودن و توانایی دریافت داده ها توسط ماشن مقصد می باشد، لذا در پروتکل TCP ابتدا یک ارتباط بین فرستنده و گیرنده ایجاد شده و پس از هماهنگی بین مبدأ و مقصد داددده ها ارسال می گردند. برای مثال فرض کنید پروسه A تمایل داشته باشد برای پروسه B برروی یک ماشین مشخص، داده هایی را ارسال نماید، قبل از ارسال داده، ماشن A یک بسته خاص به عنوان درخواست برای ارتباط به آدرس ماشین B می فرستد و منتظر می ماند. B درخواست ارتباط را ذیافت و در صورتی که آماده برقراری ارتباط باشد و یا نتواند درخواست A را برآورده کند، به ماشن A آمادگی و یا عدم آمادگی خود را اعلام می نماید. در صورتی که فرستندهٔ A در مهلت زمانی شخص پاسخ مثبت گیرنده B را دریافت نماید، آنگاه شروع به ارسال داده به گیرنده می نماید. در هنگام خاتمه مبادله داده ها نیز فرستنده

A اتمام ارسال داده های خود را به گیرنده اعلام و منتظر می ماند و در همین هنگام به دریافت داده های ارسالی از طرف B نیز ادامه می دهد تا آنکه B نیز اعلام ختم ارتباط را به فرستندهٔ A ارسال نماید.

بنابراین ارسال داده ها به صورت هماهنگ و با اطلاع قبلی انجام می گیرد. در صورتی که بهر عللی ازجمله خراب شدن خط ارتباطی ئ یا خاتمهٔ ناهنگام یکی از دو طرف فرستنده یا یگرنده در ارتباط خوشه وارد گردد، پروتکل TCP بوسیله زمان سنجی های خاص آن را کشف و مدیریت می نماید. کاستی دیگر در لایه IP تضمکین به ترتیب رسیدن داده ها و صحت آنهاست. برای برطرف کردن این نیاز پروتکل TCP به روش زیر عمل می نماید. فرض کنید پروسه فرستنده A داده های مورد نظر خود جهت ارسال را در قالب یک بسته سازماندهی نموده و سرآیند آنرا با شماره ترتیب، ترتیب بندی نموده است. فرستنده A بسته سازماندهی شده را در یک بافر نگهداری می نماید و همچنین آنرا جهت ارسال تحویل لایه IP می دهد. در همین زمان یک زمان سنج را تنظیم می نماید و نیز برای خطایابی در سمت گیرنده که ۱۶ بیتی کشف خطا را نیز بر اساس

داده های آن بسته در سرآیند می گذارد. پس از ارسال بسته، در سمت گیرنده در صورتی که گیرنده B، بسته ارسالی A را به صورت سالم (با چک کردن کدهای کشف خطا) دریافت کند، یک پیغام تصدیق (Acknowledge) یا به اختصار ACK به فرستنده می فرستد. اگر فرستنده در زمان مقرر پیغام ACK را دریافت کند، بسته ارسال شده از بافر حذف می گردد و در صورتی که به هد عللی ACK توسط فرستنده دریافت نگردد، زمان سنج تنظیم شده Time out گردیده و بسته بافر شده مجدداً ارسال می گردد. ترتیب بندی داده ها نیز بوسیله قرار دادن شماره ترتیب برای داده هاتی ارسالی و نرتب سازی بسته های دریافتی در مقصد بر اساس شماره ترتیب آن می باشد. همچنین با استفاده از این شماره می توان بسته هایی را که دوبار ارسال گردیده اند را شناسایی و یکی از آنها را حذف نمود.

به صورت خلاصه، پروتکل TCP برای پروسه هایی که از استفاده می نماید، سرویسهای زیر را تأمین می نماید.

- انتقال داده ها به صورت نهری (Stream).

این پروتکل جریان پیوسته بایتهای داده را برروی شبکه به صورت نهری منتقل می نماید و در این انتقال، برنامه های کاربردی درگیر سازماندهی داده ها در بسته های ارسالی نیستند.

- قابلیت اعتماد:

پروتکل TCP همانگونه که ذکر گردید با استفاده از شماره ترتیب (Sequence number) و نیز مکانیزم ACK برای هر بسته و نیز ارسال مجدد بسته های از بین رفته، ارتباط مطمئن را جهت ارسال داده فراهم می نماید.

- کنترل جریان

در پروتکل TCP، گیرنده در ارسال ACK به فرستنده مقدار اندازه بافر ورودی خود را به فرستنده اطلاع می دهد درنتیجه فرستنده داده ها را به صورتی که بافر گیرنده سررسیز نماید ارسال نمی کند.

- ارسال داده توسط پروسه های در حال اجرا به صورت همزمان برروی یک ماشین با استفاده از شماره پورت سوکت.

- ارتباطات منطقی: ترکیب اطلاعات سوکت، شماره ترتیب و اندازه پنجره به عنوان ارتباط منطقی شناخته می شود.
- ارتباط دوطرفه همزمان.

پروتکل TCP جریان داده همزمان را به صورت دوجهته بین فرستنده و گیرنده برقرار می کند.

قالب بسته TCP:

بسته TCP مشتکل از دو قسمت سرآیند و داده می شود. سرآیند ۲۰ بایتی بوده که قالبی ثابت دارد. بعد از این سرآیند ثابت، گزینه های سرایند نیز وجود دارد. پس از گزینه ها، در صورت لزون تا ٦٥٥١٥ بایت از داده ها وجود دارد. سرایند TCP می تواند تا ٦٠ بایت باشد.

حال فیلدهای سرآیند TCP را بررسی می نمائیم.

- شماره اعلام وصول (Ask no). این فیلد ۳۲ بیتی شماره ترتیب بایتی که فرستنده بسته منتظر دریافت آن است را تعیین می کند. بنابراین گیرنده اگر در آخرین بایت با شماره n را دریافت کرد، n+1 را به عنوان شماره اعلام وصول به فرستنده ارسال می کند.
- طول سرآیند. Header Length (HLEN): این فیلد نشادهنده طول سرایند بسته TCP را بر مبنای چهار بایت تعیین می نماید.
- بیتهای رزرو شده به تعداد ٦ عدد که جهت استفاده در آینده بلااستفاده رها شده اند.
- فیلد آدرس پورت مبدأ (Source port) که شماره ١٦ بیتی است که آدرس پورت پروسهٔ مبدأ در ارسال بسته تولید شده است.

- فیلد آدرس پورت مقصد (Destination port): این فیلد نیز شماره ۱۶ بیتی است که آدرس پورت پروسه مقصد که در ماشین مقصد بسته ارسالی را تحویل می گیرد را مشخص می نماید.
- فیلد شماره ترتیب: فیلدی ۳۲ بیتی است که شماره ترتیب آخرین بایتی را که در فیلد داده از بسته جاری قرار دارد را نشان می دهد. این شماره در هنگام برقراری ارتباط به صورت تصادفی بین فرستنده و گیرنده توافق شده و شروع می گردد.
- فیلد Checksum: این فیلد ۱۶ بیت بوده و کد کشف خطایی است که در سمت فرستنده بر اساس داده ارسالی محاسبه شده و در سمت گیرنده همان محاسبات برروی داده رسیده مجدداً تکرار می گردد و اگر نتیجه حاصل با مقدار فیلد مذکور یکسان نباشد، تمامی بسته ارسالی از بین برده شده و مجدداً درخواست ارسال آن به فرستنده ارجاع می گردد.
- فیلد Urgent Pointer: عدد واقع در این فیلد اشاره گر به موقعیت داده اضطراری واقع در بسته در حال انتقال است. داده های اضطراری می توانند بدون قطع ارتباط، داده های اضطراری را که

معمولاً شبیه وقوع وقفه ها در حین اجرای برنامه کاربردی است را به گیرنده ارسال کنند و در سکت گیرنده این داده ها از اولویت پردازشی بالاتری برخوردار خواهند بود.

- فیلد Options: سرایند می تواند حداکثر ٤٠ بایت اطلاعات دلخواه را در این بخش جای دهد. برای آنکه طول بسته ضریبی از ٤ باقی بماند از این فیلد با کدهای بی ارزش استفاده می گردد. ماکزیمم اندازه سگمنت TCP(MSS) برابر است با بیشترین حجم داده ای است که TCP به گیرنده ارسال خواهد نمود. در هنگام ایجاد یک ارتباط TCP هر یک از طرفین ارتباط MSS دریافت داده خود را به طرف مقابل اعلام می نمایند در غیر این صورت عدد پیش فرض 536 بایت به عنوان پیش فرض MSS استفاده می گردد. بنابراین با احتساب ٢٠ بایت سرآیند TCP و ٢٠ بایت به عنوان سرایند IP، در دیتاگرام IP، به صورت پیش فرض ٥٧٦ بایت قرار می گیرد. مشخصه دیگری که می تواند در فیلد Option قرار گیرد، Time stamp است که توسط فرستنده در ارسال سگمنت و جهت محاسبه round Trip time که مدت زمان ارسال

سگمنت، دریافت در سمت گیرنده و سپس دریافت ACK گیرنده در سمت فرستنده است، مشخص و در آن فیلد قرار می گیرد.

- فیلد اندازه پنجره (Windows Size): این فیلد نشاندهنده فضای خالی بافر گیرنده می باشد. گیرنده توسط این فیلد به فرستنده حداکثر اندازهٔ مقدار داده ای را که فرستنده مجاز است پس از بایت با شماره ترتیب ذکر شده در اعلام وصول (ACK) به گیرنده ارسال نماید، گوشزد می نماید. اگر این مقدار صفر باشد نشاندهنده پر بودن بافر گیرنده است.
- بیتهای پرحجم (Flag): شش بیت در بسته TCP به عنوان بیتهای پرحجم عمل می نمایند که عبارتند از

- بیت URG: در صورت یک بودن این بیت، فیلد اشاره گر اضطراری بایستی در سمت گیرنده مورد توجه قرار گیرد چرا که دارای مقداری قابل استناد و معتبر است و صفر بودن این بیت نشاندهنده آن است که گیرنده بایستی از فیلد اشاره گر اضطراری (urgent pointer) چشم پوشی نماید چرا که آن فیلد شامل مقدار معتبر و قابل استناد نیست.

- بیت syn: که در برقراری ارتباط بین فرستنده و گیرنده نقش دارد که ذکر خواهد گردید.
- بیت ACK: یک بودن این بیت نشاندهنده معتبر بودن مقدار فیلد شماره اعلام وصل می باشد. همچنین از این فیلد در هنگام برقراری ارتباط TCP نیز استفاده می گردد. بدین صورت که یک شدن بیت syn و صفر بودن بیت ACK و ارسال آن به طرف مقابل به معنای درخواست برقراری ارتباط است. در سمت گیرنده، در صورتی که گیرنده متمایل به برقراری ارتباط با فرستنده باشد، بسته ای را ارسال خواهد نمود که در آن بیت syn برابر یک و نیز بیت ACK برابر یک خواهد بود که نشاندهنده پذیرش یک ارتباط است که بر فرستنده ارسال خواهد گردید.
- بیت PSH (push): یک بودن این بیت نشاندهنده این است که فرستندهٔ اطلاعات از گیرنده تقاضا می کند که داده های موجود را بافر ننمالید بلکه آنرا تحویل برنامهٔ کاربردی خود بدهد.
- بیت RST یک شدن این بیت نشاندهنده قطع ارتباط توسط یک طرف به صورت یک طرفه و ناتمام خواهد بود و نیز این بیت می

تواند به عنوان نشاندهنده عدم پذیرش درخوسات برقراری ارتباط باشد.

- بیت FIN: هر یک از طرفین ارتباط. پس از پایان داده های خود بیت FIN را یک می کند و ارسال بیته حاوی بیت FIN یم شده نشاندهنده قطع ارسال اطلاعات توسط یک طرف است و پس از آن ارسال کننده این بسته به ارتباط گوش داده و داده های ارسالی طرف مقابل خود را تا خاتمه داده های ارسالی آن، دریافت می نماید تا زمانی که طرف مقابل نیز پس از پایان اطلاعات ارسالی خود در بسته آخر بین FIN را یک نماید و بدین ترتیب ارتباط به صورت هماهنگ و دوطرفه قطع خواهد گردید.

برقراری ارتباط TCP:

سرویسهای اتصال گرا جهت برقراری ارتباط از روش دست تکانی سه مرحله ای استفاده می نمایند. این مراحل عبارتند از:

۱- مرحله برقراری ارتباط

۲- مرحله انتقال داده ها

۳- مرحله قطع ارتباط

در مرحله برقراری ارتباط، یک ارتباط یا جلسه میان مبدأ و مقصد، تنظیم و برقرار می گردد. برای این کار در سمت سرور تابع سیستمی Listen() در حال گوش دادن به پورت خاص می باشد و مشتری نیز با تعیین آدرس IP سرور و شماره پورت پروسهٔ مقصد با تابع Connect() اقدام به ارسال درخواست اتصال خود به سرور می نماید. سپس در مرحله انتقال داده، داده ها به صورت ترتیبی برروی مسیر برقرار شده منتقل می گردد و به همان ترتیبی که ارسال گردیده اند در مقصد بازسازی می شوند. در فاز قطع ارتباط پس از آنکه ارتباط ایجاد شده دیگر جهت انتقال داده مورد نیاز نباشد، ارتباط خاتمه می یابد.

دست تکانی سه مرحله ای:

در ارتباطات مبتنی بر TCP، جهت انتقال داده بین طرفین ارتباط، بایستی ابتدا یک ارتباط بین طرفین توافق و برقرار گردد بدین معنی که هر یک از طرفین بایستی با دیگری بر روی شماره ترتیب اولیه جهت ارسال و دریافت اطلاعات همزمان گردند. شمارهٔ ترتیب که به صورت تصادفی توسط میزبان انتخاب می گردد، جهت ردیابی ترتیب

بسته های انتقالی و نیز اطمینان از عدم مفقود شدن بسته های اطلاعاتی در مسیر انتقال استفاده می گردد. همزمانی اولیه توسط مبادله واحدهایی از پیام ایجاد ارتباط که بیت کنترل (به آن syn گویند) و نیز شماره های ترتیب اولیه، انجام می گیرد. واحدهایی از پیام که بیت کنترلی syn را منتقل می کنند، syns نام گذاری می گردند. بنابراین برقراری یک ارتباط TCP نیازمند یک مکانیم مناسب جهت انتخاب شماره ترتیب اولیه و دست تکانی سه مرحله ای نسبتاً پیچیده جهت تبادل شماره های ترتیب اولیهٔ است. عملیات همزمانی نیازمند آن است که هر یک از طرفین ارتباط شماره ترتیب اولیه انتخابی خود را به طرف مقابل ارسال و سپس تأییدیه آن را تحت عنوان Acknowledgment (ACK) از طرف مقابل دریافت نمایند. و نیز هر یک از طرفین بایستی شماره ترتیب اولیه ارسالی از طرف دوم را دریافت و تأییدیه آن را به صورت ACK به طرق مقابل ارسال نمیاد. دست تکانی سه مرحله ای

۱- ارسال Syn از میزبان A به B، میزبان A توسط ارسال Syn به میزبان B، شماره ترتیب (SEQ) خود را که در اینجا X انتخاب شده، به اطلاع B می رساند.

۲- ارسال Sun از میزبان A به B، میزبان B پس از دریافت Syn، شماره ترتیب X که توسط A ارسال شده را ثبت کرده و با ارسال ACK که در آن بیت Syn، یک شده و نیز با مقدار ACK=X+1 و نیز شماره ترتیب اولیه انتخابی خود (SEQ=Y)، به میزبان A جواب می دهد. مقدار ACK=X+1 نشان دهنده آن است که به میزبان B بسته داده ای « را دریافت نموده و منتظر دریافت X+1 امین از میزبان A می باشد. به این تکنیک Forward acknowledgment گویند.

۳- ارسال ACK از میزبان A به B: میزبان A پس بسته ارسالی میزبان B را با بسته ACK ای که نشاندهنده آن است که میزبان A منتظر y+1 امین بسته ارسالی از B است (ACK=y+1) جولب می دهد.

پس از اتمام سه مرحله مذکور، اتصال برقرار گردیده است و هر یک از طرفین می توانند داده های خود را ارسال نمایند. دست تکانی سه

مرحله ای لازم است به آن علت که شماره های ترتیب در شبکه به زمان جهانی و مشخص بریا هر دو طرف وابسته نبوده و پروتکل TCP ممکن است روشهای مختلفی برای انتخاب شماره ترتیب اولیه داشته باشدذ. گیرنده اولین بسته Syn راهی جهت تشخیص آن که واحد دریافت شده، داده ای ارسالی از طرف مقابل که با تأخیر همراه بوده ندارد، جز اینکه آخرین شماره ترتیب استفاده شده در ارتباط (که ممکن است همیشه موجود نباشد) را به خاطر داشته باشد. در این صورت بایستی از مبدأ ارسال کننده درخواست تصدیق نمودن Syn دریافتی را بنماید.

Acknowledgement:

بازیابی قابل اعتماد در پروتکل TCP، تضمین می نماید که جریان داده ارسالی از یک میزبان پس از عبور از مسیرهای داده ای میانی بدون Duplicate و یا مفقود شدن تحویل ماشین مقصد می گردد. از تکنیک (Positive Acknowledgement & Retransmission) PAR عموماً جهت تضمین قابل اعتماد در پروتکل TCP استفاده می گردد. در این تکنیک مبدأ بسته داده را ارسال و یک زمان سنج را به

کار می اندازد، سپس قبل از آنکه بخش داده ای بعدی را ارسال نماید منتظر دریافت ACK بسته ارسالی می ماند. از زمان زمان سنج قبل از آنکه مبدأ، ACK بسته ارسالی خود را دریافت کند، منقضی گردد، میزبان مجدداً بسته داده خود را ارسال و دوباره زمان سنج را به کار می اندازد.

اندازه پنجره (Window Size) نشاندهنده مقدار داده ای است که در هر زمان بدون دریافت ACK از مقصد می توان به مقصد ارسال نمود و مطمئن بود که بافر ماشین مقصد سرریز نمی نماید. بنابراین شماره اندازه پنجره (WS) همیشه بایستی بزرگتر یا مساوی مقدار داده ای که میزبان می خواهد ارسال نماید. این مکانیزم را توسط شکل و دنبال کردن یک مثال بررسی می کنیم.

فرض کنید WS برای هر یک از میزبانها برابر 1 باشد، هر واحد دده مجزا با اندازه 1 بایستی قبل از آنکه واحد بعدی را ارسال نمائید تصدیق (ACK) شده باشد. (در مورد پنجره بندی TCP windowing در بخش مربوطه توضیح داده خواهد شد.)

دنباله TCP و شماره های ACK،

پروتکل TCP همانگونه که توضیح داده شد، دنباله ای از واحدهای داده به همراه ACK را جهت تبادل اطلاعات بین دو میزبان استفاده می نمایند. هرواحد بوسیله میزبان مبدأ قبل از ارسال شماره گذاری می گردد.

در ماشین مقصد TCP واحدها را به صورت پیام کامل بازسازی می نمایند. اگر در حین بازسازی واحدها شماره ترتیب (یک واحد) کم شده و یا دریافت نشده باشد، مجدداً توسط ماشین مبدأ بایستی دوباره ارسال گردد. بنابراین در مقصد اگر یک واحد به شماره ترتیب خاص دریافت نگردد و یا ACK مربوط به یک واحد دریافت شده که توسط مقصد ارسال گردید توسط فرستنده دریافت نگردد، در آن صورت زمان سنج آن واحد که در سمت فرستنده فعال بوده است منقضی گردیده و واحد داده مجدداً ارسال گردد.

خاتمه ارتباط در TCP

خاتمه ارتباط در TCP شامل دو مرحله دست تکانی بین طرفین می باشد. هر یک از طرفین ارتباط می تواند سعی در خاتمه ارتباط را آغاز کند. فرض کنید میزبان A بخواهد ارتباط بین خودش و میزبان B را با توافق طرفین خاتمه دهد. ابتدا میزبان A، حالت خود را در ارسال آخرین داده خود به حالت انتظار برای خاتمه (CLOSE WAIT) قرار می دهد. در این حالت میزبان A هم چنان داده های ارسالی از میزبان B را دریافت می نماید و ACK آنرا نیز به میزبان B ارسال می کند. در مرحله دوم میزبان B پس از آنکه دریافت اطلاعات از کاربر خود را خاتمه یافته می بیند حالت خود را به Close Prinitive برده و واحد اداری FIN را به معنای خاتمه ارتباط به میزبان 1 ارسال می کند. و بدین ترتیب ارتباط خود را خاتمه می دهد. میزبان A نیز پس از دریافت دادهٔ FIN، به ارتباط خاتمه می دهد.

Window Principle

برای نظارت بر جریان داده بین دستگاهها، پروتکل TCP از مکانیزم کنترل جریان (Flow-Control) استفده می نماید. TCP دریافتی در ارتباط، پنجره مجاز جهت ارسال داده را برای هر میزبان مشخص م کند. مقدار پنجره نشاندهندهٔ بایتهایی است که طرف مقابل در ارتباط tcp آماده دریافت آنهاست. برای مثال به ازای اندازهٔ پنجره ای معادل ۳، ماشین مبدأ می تواند سه بایت را به مقصد ارسال نماید سپس تا رسیدن ACK از سوی ماشین مقصد منتظر می ماند. اگر مقصد سه بایت را دریافت نماید، ACK دریافت آنها را به فرستنده ارسال می نماید و از آن پس فرستنده مجدداً م تواند حداکثر سه بایت دیگر را ارسال نماید. مقد ممکن است به دلایل مختلفی بایتهای ارسالی مبدأ را دریافت ننماید مثلاً به علت سرریز شدن بافرش که در آن صورت ack را به فرستنده ارسال نمی نماید و فرستنده نیز به علت آنکه ack داده ارسالی خود را از مقصد دریافت ننموده می داند که بایستی داده های ارسالی خود را مجدداً و این بار با نرخ ارسال کمتر ارسال نماید. اندازه پنجره tcp در زمان برقار بودن ارتباط متغیر است. هر ack شامل اعلان اندازه پنجره ای است که نشاندهنده تعداد بایتهایی است

که گیرنده می تواند از آن پس دریافت نماید. پروتکل tcp همچنین از کنترل برخورد (Congestion Control) جهت تعیین اندازه پنجره در زمانی که ارسال داده ها برخورد روی می دهد استفاده می کند. بدین معنی که پس از آنکه ارسال داده با برخورد مواجه گردید، الگوریتم کنترل برخورد جهت کاهش اندازه پمجره برای جلوگیری از برخوردهای بعدی استفاده می گردد. با این روش اندازه پنجره بیش از حد افزایش نیافته و در محدوده ای که از نظر بافر و قابلیت پردازشی در حد توان گیرنده است، تنظیم می گردد. افزایش یافتن اندازه پنجرهع به معنی نیاز به پردازش حجم بیشتر داده دریافتی در سمت گیرنده است. در شکل زیر

مبدأ قبل از آنکه از گیرنده ACK دریافت کند، بر اساس اندازهٔ پنجره مربوط به گیرنده، سه واحد داده را به گیرنده ارسال می نماید. گیرنده به جهت آنکه تنها اندازهٔ پنجره دو واحد را می توانسته پردازش نماید، تنها تا ۲ واحد داده را دریافت نموده است. بنابراین در جواب واحدهای داده دریافت شدهٔ خود شماره ترتیب مورد انتظار خود را معادل ۳ قرار داده و اندازه پنجره را معادل ۲ واحد تنظیم و ACK را ارسال می نماید. سپس مبدأ اندازه پنجره گیرنده را به

صورت ۲ واحد بروزرسانی کرده و در واحد داده بعدی را به گیرنده ارسال می نماید. و گیرنده نیز در جواب ACK با شماره ترتیب مورد انتظار ۵ و اندازه پنجره ۲ واحد به فرستنده ارسال می نماید.

مشکلات کنترل جریان در TCP:

بکارگیری پنجره در روند کنترل جریان در TCP دارای مشکلات اجرایی مختلفی است. این مشکلات عباتند از:

۱- مشکل بسته های کوچک (spp) the small packet problem:

در برخی از سیستمهایی که روند انتقال پیام بر اسسا داده هایی با حجم کم بوده و به صورت متناوب و به صورت بلادرنگ انتقال داده های کوچک را بین دو میزبان انجام می دهد، داده های منتقل شده نسبت به داده های سابار ناشی از عملیات قالب بندی داده ها بسیار کم بوده و بنابراین throughput شبکه بسیار امنیت می نماید. بریا مثال سیستمی را در نظر بگیرید که به محض فشرده شدن یک کلید بایستی داده خوانده شده اند صفحهٔ کلید را بلافاصله به میزبان مقصد منتقل نماید. در این صورت داده خواده شده از صفحهٔ کلید نمی تواند در بافر قرار گیرد و پس از پر شدن یا به حجم مناسب

رسیدن باز منتقل گردد. بنابراین میزبان مبدأ یک بایت داده خوانده شده را در پکت حداقل ٤٠ بایتی قرار داده و آنرا به میزبان مقصد ارسال می نماید. از طرفی در سمت گیرنده میزبان پس از بازیابی داده از بسته رسیده، اقدام به ارسال ACK به فرستنده می نماید همچنین گاهاً برخی از برنامه ها داده ارسالی را به صورت Echo به عنوان ACK به فرستنده ارسال می نمایند. مشکل SPP به صورت جدی باعث کاهش کارایی شبکه می گردد.

برای حل مشکل SPP الگوریتم Nagle به صورت زیر مطرح گردیده است: در این الگوریتم به جای ارسال تک تک داده های تولید شده به مقصد، پس از ارسال اولین بایت و دریافت ACK مربوط به آن تمامی داده های موجود در بافر به یک باره، دسته بندی و ارسال می گردد.

۲- شکل پنجره نانجرد Silly Window Syndrom (SWS):

این مشکل زمانی ایجاد می گردد که فرستنده در ارتباط TCP بسته های داده ای با اندازه زیاد را به گیرنده ارسال می نماید اما در سمت گیرنده بعلت سرعت پردازش پائین و یا هر وسیلهٔ دیگر، گیرنده در هر زمان تنها یک بایت از اطلاعات رسیده را از بافر خوانده و ACK

مربوط به اعلام به روزرسانی بافر خود را به فرستنده ارسال می کند. بدین ترتیب فرستنده نیز متوجه می گردد که گیرنده در هر زمان تنها می تواند یک بایت را بخواند بنابراین فرستنده نیز تنها در هر زمان یک بایت را ارسال می نماید و این ارسال تک تک داده ها تا زمانی که همه اطلاعات ارسال گردد ادامه می یابد. مشکل SWS. بوسیله راه حل Clark مرتفع می گردد. در این روش برنامه هایی که از بافر TCP داده های مربوط خود را می خوانند مجلور می گردند تا داده های خود را به صورت تک بایت نخوانند و تنها زمانی می توانند از بافر درخواست خواندن اطلاعات را نمایند که این درخواست مربوط به خواندن حجم زیادی از داده های بافر شد. در این گونه موارد فرستنده نیز می تواند بسته های ارسالی با داده های با حجم کم را به گیرنده ارسال نماید.

۳- انقضاء زمانی ارسال Retransmition Timeout (RT).

همانگونه که ذکر گردید زمانی که TCP واحد داده ای را به گیرنده ارسال می کند، زمان سنجی را برای آن واجد آغاز می نماید و در صورتی که بعد از زمان خاصی، ACK مربوط به آن واحد از گیرنده

دریافت نگردید، زمان سنج از نظر زمانی منقضی (Time Out) می گردد و فرستنده واحد داده را ارسال مجدد می نماید. به صورت کلی پس از آغاز به کار زمان سنج حالات مختلفی ممکن است به وجود آید.

۱- ممکن است ACK مربوط به داده ارسال شده به مقصد در سمت فرستنده دریافت گردد ولی در هنگام دریافت آن زمان سنج هنوز منقضی نگردیده و یا حتی زمان سنج از بین رفته باشد که این حالت اشکالی را در ارتباط ایجاد نم نماید.

۲- اگر زمان انقضاء ارسال مجدد واحد داده (RTO) Retransmotion Time Out قبل از رسیدن ACK مربوط به آن از گیرنده به سر اید، آن واحد داده مجدداً به گیرنده ارسال و زمان سنج آن مجدداً کار خود را از ابتدا آغاز می نماید.

۳- اگر واحد داده ارسالی از فرستنده توسط گیرنده ACK نگردد، نشان دهنده آن است که داده ارسالی و یا ACK آن در میان راه، مفقود و یا خراب گردیده اند که در این صورت نز داده ارسال مجدد و زمان سنج دوباره تنظیم می گردد.

زمان RTT (Round-Trip Time) به مدت زمانی اطلاق می گردد که واحد داده از فرستنده ارسال تا ACK آن توسط فرستنده دریافت گردد. این زمان متأثر از عواملی مانند تعداد برخوردها در شبکه انتقالی و نیز تعداد آن توسط فرستنده دریافت گردد. این زمان متأثر از عواملی مانند تعداد برخوردها در شبکه انتقالی و نیز تعداد گرههای میانی بین فرستنده و گیرنده (راهگزین ها Rowter) می باشد. در تعیین زمان RTO بایستی به زمان RTT نیز توجه گردد. لذا معمولاً با تخمین زدن نرمال RTT می توان زمان مناسب برای RTO را به صورت مناسبی بدست آمد. به علت آنکه شبکه از نظر بار گرافیکی و نیز تعمیرات ساختاری بایستی در مقابل تغییرات قابل انعطاف باشد لذا تعیین زمان RTT بایستی به صورت پویا باشد و نیز بایستی به نوعی اثر تغییرات لحظه ای در زمان RTT را نیز تا حد ممکن کم کرد. بنابراین RTT هموار (smooth RTT (SRTT)) مقدار ATT نرمال مورد نیاز برای تعیین RTO درنظر می گیریم. بنا به تعریف.

SRTT=α×STT+(1-α)×[اندازه گیری شده جاریRTT]

وزن SRTT بر اساس RTTهای قبلی با ضریب وزنی α که اغلب برابر ۰/۸۷۵ است به اضافه RTT جاری مربوط به آخرین بسته ACK شده بین فرستنده و گیرنده درنظر می گیرند.

آنگاه RTO را که قاعدتاً بایستی از RTT بزرگتر باشد تا زمان سنج قبل از رسیدن ACK منقضی نگردد، را به صورت زیر تعریف می کنند.

$$RTO=\beta\times SRTT$$

که ضریب β مقداری بزرگتر از 1 بایستی باشد.

کنترل برخورد در TCP:

زمانی که مسیریابهای میانی در ارتباطات TCP، یک مسیریاب سرعت دریافت بسته های اطلاعاتی جهت مسیریابی از سرعت پردازش بسته دریافتی و پردازش آن و تعیین مسیر و سپس ارسال آن در مسیر تعیین شده بهتر باشد باعث می گردد صف بسته های دریافتی در مسیریاب طولانی تر گردد و اگر از حد ظرفیت مسیریاب بیشتر گردد، بسته های رسیده، توسط مسیریاب از بین می رود که به این حالت network congestion گویند. از طرفی مسیریاب پس

از خراب نمودن بسته دریافتی توسط پیامهای کنترلی خراب شدن بسته را به اطلاع فرستنده می رساند. فرستنده نیز بسته مفقود شده را مجدداً ارسال می نماید که این ارسالهای مکرر باعث می گردد که شبکه عملاً از کارایی خود باز بماند که به آن Congestion Collapse گویند زمانی اتفاق می افتد که شبکه در حالت استفاده صددرصد می باشد ولی تعداد کمی از بسته های ارسالی از فرستنده ها به گیرنده مربوطه می رسند.

بنا بر آنچه گفته شد، بسته ارسال شده توسط فرستنده از شبکه های مختلف و مسیریابهای مختلف گذشته و به مقصد می رسد. هر مسیریاب دارای بافر ورودی بوده که در آن بسته های رسیده از مبدأهای مختلف به مسیریاب ذخیره می گردد. مسیریاب بسته ها را از بافر خوانده، آنرا پردازش و سپس آنرا به مسیر ارتباطی مربوطه خود منتقل می نماید.

در مسیرهایی که یک بسته طی می کند تا به مقصد برسد، مسیری که دارای کمترین پهنای باند و یا کیفیت است، تعیین کننده سرعت ارسال بسته بین فرستنده و گیرنده می باشد. (bottleneck). بنابراین یک شمای مدیریت برخورد جهت جلوگیری فرستنده از ارسال تعداد

بسته های بیشتر از ظرفیت دریافت و تحویل بسته از مقصد به مبدأ مورد نیاز است. بنابراین علاوه بر گیرنده، خود شبکه نیز موجودیتی است که در تعیین اندازه پنجره فرستنده بایستی تأثیرگذار باشد. از اینرو مقدار صحیح پنجره ارسال در سمت فرستنده را برابر حداقل پنجره اعلام شده توسط گیرنده به فرستنده و نیز اندازه پنجره برخورد (Network Congestion Window) تعریف می کنیم. برای تعیین اندازه پنجره برخورد (Congstion Window)، TCP از تکنیکهای زیر استفاده می نماید.

۱- آغاز آهسته (Slow start):

در این تکنیک اندازه اولیه پنجره برخورد (Cwnd) در آغاز برابر ۱ واحد داده انتخاب می گردد. بنابراین فرستنده یک واحد داده را ارسال نموده و ACK مربوط به آن را از طرف گیرنده دریافت می نماید. پس از دریافت ACK، فرستنده اندازه پنجره خود را یک واحد افزایش می دهد و این بار، دو واحد داده را ارسال می نماید. پس از دریافت ACK آنها، این بار 4 واحد داده را ارسال می کند. این

افزایش تعداد واحدهای داده به صورت توان دو ادامه می یابد. پس در هر ارسال (RTT)، اندازه پنجره Cwnd دو برابر می شود.

در این تکنیک در صورتی که یک پکت از بین برود و یا ACK آن به موقع دریافت نگردد، اندازه پنجره دوباره برابر ۱ می گردد و مراحل قبل دوباره تکرار می گردد. همانگونه که مشخص است افزایش اندازه پنجره در این تکنیک بسیار سریع و به صورت نمایی است و نیز ارتباط بسیار ناپایدار است و علت آن نیز این است که اگر یک بسته در مسیر ارتباطی از بین برود، اندازه پنجره برابر یک می گردد.

واضح است که در این روش اندازه پنجره به through put بهینه همگرا نیست.

۲- پرهیز از برخورد (Congestion Avoidance):

در این روش اندازه پنجره برخورد (Cwnd) تا مقدار threshould (برپا آغاز این مقدار برابر 64k انتخاب می گردد) به صورت نمایی و همانند موش قبل رشد می نماید. پس از آنکه اندازه پنجره به مقدار

threshould رسید، از آن پس در ازای رسیدن ACK هیا مربوط به واحدهای ارسال شده، تعداد واحدهای ارسالی (اندازه پنجره برخورد Cwnd) یک واحد اضافه می گردد. اگر فرستنده ACK مربوط به داده های ارسال خود را دریافت ننماید و زمان سنج از RTO بیشتر شود، فرستنده استنباط می کند که واحد داده توسط مسیریابهای میانی از بین رفته که این نشان دهنده بروز برخورد در شبکه است. بایستی توجه داشت که با توجه به اینکه شبکه های امروزی اغلب full switch هستند (nise free) لذا احتمال مفقود شدن بسته بسیار بیشتر از خراب شدن آن است. در این روش پس از آنکه بروز برخورد کشف گردید مقدار threshold برابر نصف آخرین اندازه پنجره برخورد (cwnd) قرار داده و اندازه پنجره همانند روش قبل از تعداد یک واحد آغاز می گردد.

۳- ارسال مجدد سریع TCP (Fast Retransmition):

دو روش قبل throughput بالا ندارند و به throughput بهینه نیز همگرا نیستند. روش دیگر کنترل برخورد در TCP آن است که از

تأخیرهای طولانی برای انتظار انقضاء زمان زمان سنج خودداری نمائیم. هماهنگونه که در شکل می بینید رسیدن سه ACK تکراری به صورت پیوسته نشاندهنده آن است که واحد داده ارسالی مفقود گردیده است زیرا ACK های تکراری تنها بر اساس دریافت واحدهای داده جدید تولید شده لذا برای واحد داده مذکور بجای آنکه منتظر منقضی شدن زمان سنج آن باشد، بلافاصله آنرا مجدداً ارسال می کند.

۴- نیازهای سریع TCP (Fast Recovery)،

در این روش در آغاز همانند روش اول، رشد اندازه پنجره برخورد به صورت نمایی بوده و در هر برخورد، ssthreshould برابر نصف مینیمم اندازه پنجره گیرنده و اندازه پتجره برخورد تعیین می گردد و مقدار اندازه پنجره برخورد برابر با مقدار ssthre should به اضافه سه برابر اندازه واحد داده قرار داده می شود. با رسیدن هر ACK تکراری اندازه پنجره برابر اندازه یک واحد داده اضافه گردیده و ارسال داده ها انجام می گردد. با رسیدن ACK غیرتکراری بعدی به فرستنده، اندازه پنجره برخورد برابر ssthreshold می گردد.

User Daragram Protocol:

یکی از پروتکلهای لایه انتقال، پروتکل UDP می باشد. این پروتکل ارتباطی غیراتصالگرا و غیرقابل اطمینان ایجاد می نماید. از این پروتکل جهت ارسال پیامها بین میزبانها استفاده می گردد. در این لایه هیچ نظارتی برروی تحویل داده برای واحدهای داده ارسالی به مقصد وجود ندارد. بنابراین UDP برای ایجاد قابلیت اعتماد، از پروتکلهای لایه برنامه هیا کاربردی استفاده می کند و بنابراین UDP برای برنامه های کاربردی که روند بازیابی خطا را در خودشان مدیریت می نمایند، طراحی گردیده است. این پروتکل در عوض قابلیت اعتماد، سرعت ارتباطی بالاتر را عرضه می کند. این پروتکل کنترل جریان را مدیریت ننموده و پیامهای رسیده را بازسازی مجدد نمی نماید و بنابراین بازیابی مرتب بسته ها را در نقصد تضمین نمی نماید.

سرعت بالای این پروتکل به لحاظ آن است که پردازش بسته UDP بویژه در مبدأ بسیار آسان می باشد.

قالب سرآیند UPD

پروتکل UPD داده ها را به صورت غیرقابل اعتماد بین میزبانها منتقل می نمایند. در شکل زیر ساختار سرآیند بسته UDP نشان داده شده است. همانگونه مشخص است. این سرآیند فاقد فیلد شماره ترتیب و یا ACK می باشد. طول سرآیند UDP معمولاً 64 بیت می باشد. فیلدهای سرایندذ UDP عبارتند از:

- پورت مبدأ (Source Port): این فیلد حاوی شماره پورت ارسال کننده در مبدأ می باشد.
- پورت مقصد (Sedtination Port): حاوی شماره پورت مربوطه در میزبان مقصد است.
- طول (Length): این فیلد، طول سرآیند UPD و داده آنرا مشخص می نماید.
- Checksum: این فیلد ۱۶ بینی، حاوی Checksum محاسبه شده از سرآیند داده بسته UDP می باشد.
- داده (DATA): این فیلد به صورت اختیاری بوده و حاوی فیلدهای مربوط به لایه های بالاتر از لایه انتقال می بادش.

پروتکلهایی که از UPD جهت لایه انتقال خود استفاده می کنند متنوعند. مثالهایی از آنها عبارتند از:

- (TFTP) Trivial File Transfer Protocol
- (SNMP) Simple network management
- (NFS) Network File System
- (DNS) Domain name system